

BIBLIOTECA PRACTICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

TALLER DE INFORMATICA

PROGRAMAS VALIDOS PARA AMSTRAD,

IBM, SPECTRUM, COMMODORE Y MSX



¿UN TALLER DE INFORMATICA?

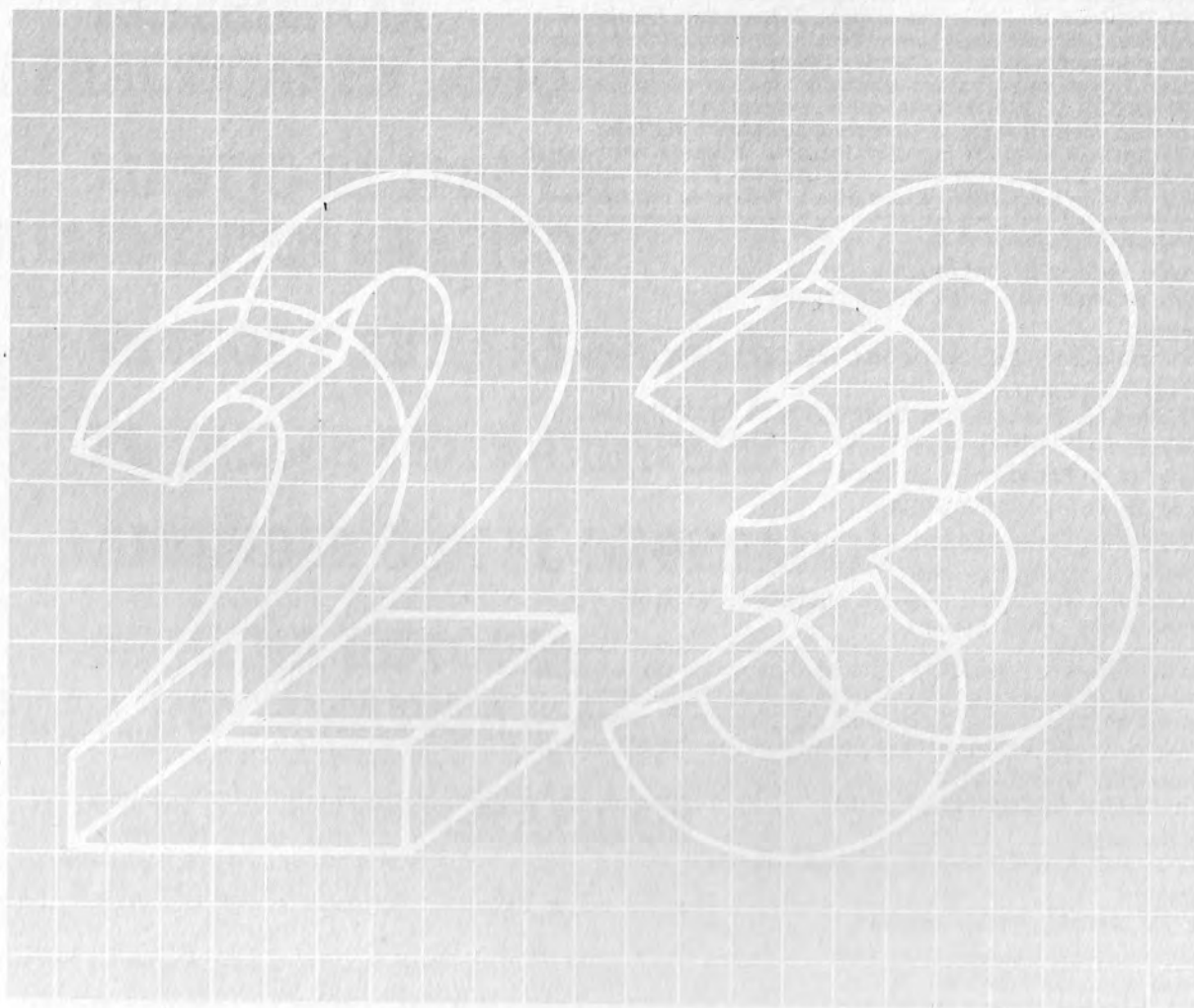
ATREVASE A HACER BRICOLAGE CON SU O.P.

EDICIONES SIGLO CULTURAL

BIBLIOTECA PRACTICA TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD,
IBM, SPECTRUM, COMMODORE Y MSX



EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S. A.

Director-editor:

RICARDO ESPAÑOL CRESPO.

Gerente:

ANTONIO G. CUERPO.

Directora de producción:

MARIA LUISA SUAREZ PEREZ.

Director de la colección:

JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño:

BRAVO-LOFISH.

Maquetación:

D. S. M.

Dibujos:

JOSE OCHOA

Tomo 23. «Experiencias y prácticas en logo», Carlos Albert, Técnico de Informática; Adoración Llena Jubero, Técnico de Informática. «Manejo de sprites y elementos gráficos», «Trucos y rutinas básicas», Francisco Morales. Técnico de Informática. «Aprende con el ordenador», AULA DE INFORMATICA APLICADA: Jesús Bocho, Licenciado en Informática; Alejandro Marcos, Licenciado en Química. «El Taller del Hardware», Carlos Rey, Ingeniero Industrial. «Pequeña historia de la Informática», «Temas monográficos de vanguardia», «Terminología», «Vocabulario de Informática», Blanca Arbizu, Técnico de Informática.

Ediciones Siglo Cultural, S. A.

Dirección, redacción y administración:

Pedro Teixeira, 8-2.º. Teléf. 810 52 13. 28020 Madrid.

Publicidad:

Gofar Publicidad, S. A. Benito de Castro, 12 bis. 28028 Madrid.

Distribución en España:

COEDIS, S. A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.

Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:

CADE, S.R.L. Pasaje Sud América, 1532. Teléf.: 21 24 64.

Buenos Aires - 1.290. Argentina.

Todos los derechos reservados. Este libro no puede ser, en parte o totalmente, reproducido, memorizado en sistemas de archivo, o transmitido en cualquier forma o medio, electrónico, mecánico, fotocopia o cualquier otro, sin la previa autorización del editor.

ISBN del tomo: 84-7688-131-2

ISBN de la obra: 84-7688-047-2

Fotocomposición:

ARTECOMP, S. A. Albarracín, 50. 28037 Madrid.

Imprime:

MATEU CROMO. PINTO (Madrid).

© Ediciones Siglo Cultural, S. A., 1986

Depósito legal: M-43.185-1986

Printed in Spain - Impreso en España.

Suscripciones y números atrasados:

Ediciones Siglo Cultural, S. A.

Pedro Teixeira, 8-2.º. Teléf. 810 52 13. 28020 Madrid.

Julio, 1987

P.V.P. Canarias: 355.—Ptas.

INDICE

■ EXPERIENCIA Y PRACTICAS EN LOGO	5
■ MANEJO DE SPRITES Y ELEMENTOS GRAFICOS	15
■ TRUCOS Y RUTINAS BASICAS	23
■ EL TALLER DEL HARDWARE	31
■ APRENDER CON EL ORDENADOR	38
■ PEQUEÑA HISTORIA DE LA INFORMATICA	48
■ TEMAS MONOGRAFICOS DE VANGUARDIA	53
■ TERMINOLOGIA	57
■ VOCABULARIO DE INFORMATICA	59

EXPERIENCIA Y PRACTICAS EN LOGO

Palabras



NA palabra es un conjunto de caracteres escritos uno a continuación del otro sin dejar un espacio en blanco entre ellos.

Una palabra puede estar formada por un número indeterminado de carac-

terres, pero al menos debe contener uno para que lo sea.

Los caracteres que componen una palabra pueden ser:

- Letras mayúsculas: A-Z.
- Letras minúsculas: a-z.
- Números: 0-9.
- Caracteres especiales: !, #, %, ...

Cada uno de los elementos que forma una palabra se denomina elemento.

El Logo considera como elemento a un conjunto de caracteres con o sin significado gramatical, como, por ejemplo:

- HOLA
- H9L5
- :?H3PP
- H
- 3H
- 7

Una palabra que contiene un solo elemento se llama carácter.

Una palabra que no contiene elemento alguno se llama palabra vacía. En realidad,

esta palabra sí está formada por al menos un carácter, ya que el espacio en blanco se considera carácter.

Las palabras están delimitadas por caracteres separadores (normalmente el espacio en blanco), pudiendo utilizar otros, como son los paréntesis y los corchetes.

Una palabra no puede contener algunos caracteres, ya que poseen funciones determinadas. Estos caracteres formarán una palabra, si son el único elemento que la constituyen.

Existen otros caracteres que tampoco pueden formar parte de una palabra, a no ser que sean el único elemento que la forman. Estos caracteres son los operadores aritméticos.

Una palabra Logo será, por ejemplo:

- +

pero nunca podrá ser:

- H+8

Una vez que conocemos lo que es una palabra Logo, veamos cómo y para qué podemos utilizarla.

■ Escribiendo palabras

Antes de comenzar a ver la manera de escribir palabras con el Logo, conozcamos con mayor profundidad la primitiva ESCRIBE.

Esta primitiva podemos escribirla bien sin abreviar:

? ESCRIBE objeto

Una palabra es un conjunto de caracteres escritos uno a continuación de otro sin dejar ningún espacio entre ambos.

EXPERIENCIA Y PRACTICAS EN LOGO

o bien en su forma abreviada:

? ES objeto

En la versión del Logo para del SPEC-TRUM:

? ESC objeto

Esta orden nos presenta en pantalla el objeto que se especifica y a continuación salta de línea.

Si, por ejemplo, queremos obtener en la pantalla el valor que contiene una determinada variable, tenemos que indicarlo de la siguiente manera:

— ESCRIBE :V

y el valor de la variable :V se mostrará en la pantalla, siempre y cuando dicha variable esté definida e inicializada.

Vamos a ver cómo podemos escribir palabras cuando trabajamos con el Logo.

El Logo reconoce que lo que se va a escribir es una palabra cuando se encuentra al objeto precedido de unas comillas.

— ? ESCRIBE "objeto

por ejemplo:

? ESCRIBE "HOLA

en la pantalla aparece:

HOLA

Si, por ejemplo, introducimos:

? ESCRIBE "COMO ESCRIBE "ESTAS

en la pantalla aparece:

COMO
ESTAS

Cada orden ESCRIBE realiza un salto de línea después de presentar en la pantalla el objeto.

Si introducimos:

? ESCRIBE "COMO ESTAS

en la pantalla aparece:

COMO

y a continuación un mensaje que dice:

NO SE QUE HACER CON ESTAS

Cuando encuentra un espacio en blanco detecta que la palabra ha terminado.

Los números, que también pueden formar una palabra, tiene un tratamiento especial. No es necesario indicar las comillas para identificarlo con una palabra:

? ESCRIBE 123456

y en la pantalla aparecerá:

123456

También es perfectamente válido introducir:

? ESCRIBE "123456

y obtenemos el mismo resultado.

Una palabra compuesta exclusivamente de números puede ir escrita sin comillas, pero en el momento que contenga un carácter que no sea numérico ha de llevarlas.

Si introducimos:

? ESCRIBE 12345T

recibimos el mensaje:

NO SE COMO HACER PARA 1234T

La forma correcta sería:

? ESCRIBE "12345T

Como hemos 'dicho anteriormente, hay algunos caracteres que no pueden formar parte de una palabra, a no ser que sean el único elemento que la forman.

Por ejemplo:

? ESCRIBE "+

y en la pantalla obtenemos:

+

? ESCRIBE "H+

y en la pantalla aparece:

H

y a continuación un mensaje de error diciendo

FALTAN DATOS PARA +

Si se utiliza un operador aritmético significa que se está solicitando que se escriba el resultado de una expresión aritmética, por ejemplo:

Hay caracteres que no pueden formar parte de una palabra.

? ESCRIBE 1+3

y en la pantalla aparece el resultado de la expresión:

4

y no la palabra:

1+3

Si se utilizan las comillas:

? ESCRIBE "1+3

el resultado sería el mismo:

4

¿Pero qué ocurre si lo que queremos escribir es una palabra de este tipo?

Para solucionar esto tenemos que utilizar unos caracteres especiales.

Si deseamos escribir, por ejemplo, la palabra:

5+7

utilizamos el carácter dólar (\$):

? ESCRIBE "5\$+7 "

y en la pantalla obtenemos:

5+7

y no el resultado de la expresión.

En el Logo de los MSX, en lugar del \$, utilizamos la /.

Estos caracteres nos permiten escribir palabras que contienen caracteres separadores u operadores.

Este carácter hay que colocarlo justamente delante del carácter separador y operador.

Gracias a estos caracteres podemos escribir palabras que estén separadas por un espacio en blanco.

Por ejemplo, si deseamos escribir:

COMO ESTAS

lo indicamos:

? ESCRIBE "COMO\$ ESTAS

y en la pantalla aparece:

COMO ESTAS

De esta forma podemos escribir palabras seguidas para formar frases:

? ESCRIBE "¿COMO\$ TE\$ LLAMAS?

En la pantalla aparecerá:

¿COMO TE LLAMAS?

El signo dólar tiene que indicarse en todos los sitios donde se desee que un carácter separador no actúe como tal.

Para escribir una palabra vacía, lo indicamos:

? ESCRIBE "

y lo que obtenemos es una línea en blanco:

? ESCRIBE "COMO ESCRIBE " ESCRIBE "ESTAS

en la pantalla aparece:

COMO

ESTAS

■ Variables y palabras

Una palabra puede asignarse como valor a una variable:

? HAZ "V "MESA

La palabra MESA se le asigna a la variable V:

Observa que para asignar una palabra a una variable tenemos que escribir unas comillas delante de ella.

En cualquier momento podemos conocer el contenido de la variable con sólo ordenar que se escriba:

? ESCRIBE :V

y en la pantalla aparece:

MESA

Si el valor que se le asigna a una variable es una palabra compuesta únicamente de caracteres numéricos:

? HAZ "V "243

el contenido de la variable :V es el mismo que si se hubiese introducido:

? HAZ "V 243

Poco a poco iremos aprendiendo otras posibilidades que nos ofrece el Logo para el manejo de palabras.

El espacio en blanco, los paréntesis y los corchetes son caracteres separadores.

EXPERIENCIA Y PRACTICAS EN LOGO

Manejo de palabras

PALABRA? obj

Esta primitiva nos informa si **obj** es una palabra o no.

Si **obj** es una palabra, la orden en cuestión devuelve CIERTO; si no, devuelve FALSO.

Es preciso tener en cuenta que en LOGO los números son palabras especiales que no precisan las comillas. Si el **obj** que acompaña a PALABRA? es un número, el resultado de la pregunta es CIERTO.

Por el momento, y hasta que no empecemos a utilizar LISTAS, no podemos comprobar la utilidad de esta orden, pero sí saber que existe.

Veamos con algunos ejemplos cómo utilizar esta primitiva.

? PALABRA? "HOLA
NO SE QUE DEBE HACER CON CIERTO

No hemos indicado qué queremos hacer con la información que devuelve esta orden.

? ESCRIBE PALABRA? "HOLA
CIERTO

Ahora el Logo ya sabe qué debe hacer con CIERTO, debe escribirlo en la pantalla.

? HAZ "HOLA "SALUDO
? ESCRIBE PALABRA? :HOLA
CIERTO

El contenido de la variable HOLA es una palabra.

? ESCRIBE PALABRA? HOLA
NO SE COMO DEBO HACER PARA HOLA

Si HOLA es una palabra, faltan las comillas delante; si es una variable, faltan los dos puntos delante.

? ESCRIBE PALABRA? 123
CIERTO

En este caso **obj** es un número, y como número puede ir sin las comillas.

? ESCRIBE PALABRA? "123
CIERTO

El objeto en este caso es una palabra, pues hemos colocado delante unas comillas.

NOTA: En la versión, SPECTRUM esta orden equivale a:

? ES.PALABRA? "obj

Otra primitiva para manejar palabras es:

PALABRA

Convierte en una sola palabra la serie de palabras que indiquemos detrás de la primitiva.

La sintaxis de la orden en el caso de unir dos palabras es:

PALABRA "palabra1 "palabra2

Es preciso indicarle al Logo qué debe hacer con la palabra resultante, si no lo hacemos se producirá un error.

? PALABRA "SEMANA "L
NO SE QUE DEBO HACER CON SEMANAL

Indicamos, por ejemplo, que escriba la palabra resultante en la pantalla.

? ESCRIBE PALABRA "SEMANA "L
SEMAMAL

Especial cuidado tendremos con los caracteres separadores; son causa corriente de errores.

? ESCRIBE PALABRA "SEMANA " L
SEMAMAL
NO SE QUE DEBO HACER CON L

El espacio en blanco es un carácter separador; por tanto, la letra L está suelta.

Sí es correcto lo siguiente:

? ESCRIBE PALABRA "SEMANA "
SEMAMAL

Unimos SEMANA con la palabra vacía.

Siempre que utilicemos la primitiva PALABRA debemos tener en cuenta dos cosas:

1: Si queremos unir más de dos palabras, la orden PALABRA con los datos a unir deben ir entre paréntesis.

? ESCRIBE (PALABRA "SUPER "CALI-
FRA "GILISTICO "ESPALIDOSO)

*A una variable se le puede asignar una palabra.
Para escribir varias palabras tienes que utilizar
caracteres especiales.*

En la pantalla aparecerá la famosa palabra:

SUPERCALIFRAGILISTICOESPIALIDOSO

2: Si la orden PALABRA sólo tiene un dato y detrás hay más órdenes, la primitiva y su dato deben ir entre paréntesis.

```
? ESCRIBE (PALABRA "HOLA) HAZ "C
10
```

■ Trabajando con la primitiva palabra

Utilizando esta orden formamos el plural y el singular de una palabra determinada, por ejemplo, de SEMANAL.

```
? PARA PALA
> HAZ "C LEECAR
> SI :C = 1 [ HAZ "A "L ] [ HAZ "A "LES ]
    > ESCRIBE PALABRA "SE-
      MANA :A
> FIN
```

El procedimiento solicita un carácter del exterior; si pulsamos el 1, indicamos que queremos el singular, y si pulsamos el 2, el plural.

Podemos hacerlo de otra forma que gramaticalmente es más correcta.

```
? PARA PALA
> HAZ "C LEECAR
> SI :C = 1 [ HAZ "A " ] [ HAZ "A "ES ]
> ESCRIBE PALABRA SEMANAL :A
> FIN
```

Si se pulsa el 1 a la palabra SEMANAL, no se le añade nada, y si se pulsa un 2, se le añade la terminación ES.

■ Leyendo datos del teclado

Ya hemos visto cómo se lee un carácter del teclado; veamos ahora cómo podemos leer varios caracteres al mismo tiempo.

Algunas versiones de Logo disponen de una orden para leer un número determinado de caracteres, esta orden es:

LEECARS n

que abreviada es:

LCS n

Junto a la orden, indicamos el número de caracteres que queremos leer.

Los caracteres que tecleamos no aparecen en pantalla, salvo que lo especifiquemos.

Para leer un solo carácter indicamos un uno.

```
? ESCRIBE LCS 1
```

si pulsamos la tecla (L), en la pantalla aparecerá la letra L.

Para leer cinco caracteres indicamos un cinco.

```
? ESCRIBE LCS 5
```

Los caracteres leídos del teclado podemos guardarlos en una variable, por ejemplo:

```
? HAZ "A LCS 4
? ESCRIBE :A
```

El ordenador aceptaría cuatro caracteres del teclado los almacenaría en la variable A y luego los escribiría.

■ Cuadro resumen

— PALABRA? "obj

Devuelve CIERTO si **obj** es una palabra; si no, devuelve FALSO.

En la versión del SPECTRUM es:

ES.PALABRA "obj

— PALABRA "palabra1 "palabra2

Permite unir varias palabras en una.

La orden y sus datos irán entre paréntesis cuando:

— Queremos unir más de dos palabras.

— La orden sólo contiene un dato y detrás hay otra orden Logo.

— LEECARS n

LCS n

Nos permite leer **n** caracteres del teclado al mismo tiempo.

Sólo disponible en algunas versiones del LOGO para los PC-Compatibles.

Si quieres escribir una línea en blanco introduce es ".

EXPERIENCIA Y PRACTICAS EN LOGO

Ejercicios

1. Define un procedimiento que lea cualquier palabra que tecleamos y que no contenga la orden LEEPALABRA.
Partiendo del procedimiento anterior, define otro para leer palabras con un número determinado de caracteres.
2. Utilizando procedimientos recursivos intenta realizar el siguiente dibujo.

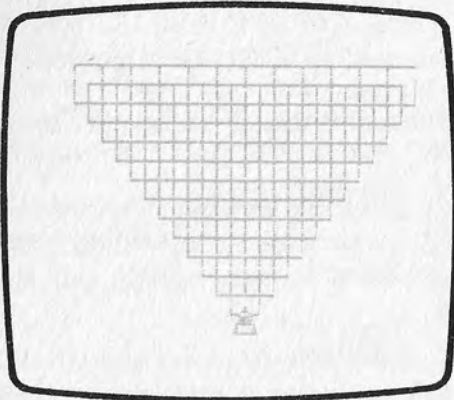


Fig. 1.

Define los procedimientos para que el número de niveles del dibujo sea variable.

3. Intenta realizar el siguiente dibujo:

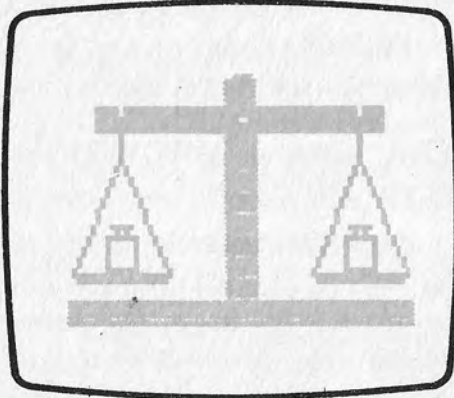


Fig. 2.

4. Define los procedimientos necesarios para calcular la aceleración de un sistema de dos masas como el que se muestra en la figura 3.

Te puede servir de ayuda saber que:

$$m_1 \cdot g - T = m_1 \cdot a$$

$$T - r \cdot m_2 \cdot g = m_2 \cdot a$$

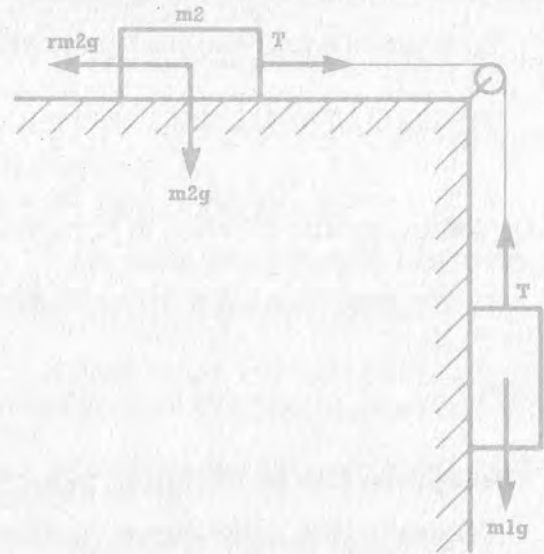


Fig. 3.

donde m_1 y m_2 son las masas de cada cuerpo, g la gravedad (9,8), T la tensión de la cuerda r el coeficiente de rozamiento y a la aceleración.

De aquí podemos deducir la fórmula de la aceleración:

$$\frac{(m_1 - r \cdot m_2) \cdot 9,8}{m_1 + m_2}$$

5. ¿Son correctas las siguientes órdenes?:

- ? ESCRIBE "P2,E%E
- ? ESCRIBE "DONDE ESTAN LAS LLAVES
- ? PARA A
 - > HAZ "V LUIS
 - > HAZ "W "110
 - > AV :V
 - > RE :W
 - > FIN
- ? PARA E2
 - > ESCRIBE PALABRA? "
 - > FIN
- ? PARA E3
 - > HAZ "B LEECAR
 - > SI ASCII :B = 13 [ALTO]
 - > ESCRIBE PALABRA :B
 - > E3
 - > FIN
- ? HAZ "C (PALABRA "FAL "SO)
- ? ESCRIBE :C

Una palabra con un solo carácter se denomina carácter.

Solución a los ejercicios

1:

Primero definimos dos procedimientos que simulen la primitiva LEEPALABRA.

El procedimiento INI inicializa la variable B, llama al procedimiento DATO y cuando termina la ejecución de éste escribe el contenido de la variable B.

```
? PARA INI
> HAZ "B "
> DATO
> ESCRIBE :B
> FIN
```

El procedimiento DATO lee, mientras no pulsemos la tecla RETURN, caracteres del teclado y los acumula, a través de la primitiva PALABRA, en la variable B.

```
? PARA DATO
> HAZ "A LEECAR
> HAZ "I ASCII :A
> HAZ "B PALABRA :B :A
> SI :I = 13 [ ALTO ] [ DATO ]
> FIN
```

Una vez introducidos estos dos procedimientos, ejecútalos tecleando:

```
? INI
```

y empieza a teclear la palabra que desees escribir.

Basándonos en los dos procedimientos anteriores, definimos los que nos servirán para leer palabras con un determinado número de caracteres.

Para ello definimos un procedimiento de una entrada, con la que indicaremos el número de caracteres que debe tener la palabra.

Y como en el caso anterior, inicializamos la variable B, ejecutamos el el procedimiento DATO y escribimos el valor de B.

```
? PARA PL :N
> HAZ "B "
> DATO :N
> ESCRIBE :B
> FIN
```

Al procedimiento DATO se le pasa el valor de la variable N, que es el tope de caracteres a leer.

Este procedimiento se ejecuta hasta que el valor de la variable N sea 1.

```
? PARA DATO :N
> HAZ "A LEECAR
> HAZ "B PALABRA :B :A
> SI :N = 1 [ ALTO ] [ DATO :N - 1 ]
> FIN
```

Si queremos leer del teclado una palabra de cinco caracteres ejecutamos el procedimiento PL indicando como dato del 5.

```
? PL 5
```

El ordenador aceptará cinco caracteres del teclado.

2:

Para realizar el dibujo definimos dos procedimientos de una entrada y recursivos.

El dibujo consta de N niveles con una serie de N rectángulos en cada uno.

Con el primero (DIBU) situamos a la Tortuga para dibujar la serie de rectángulos de cada nivel y ejecutamos MONTE. El procedimiento se ejecuta hasta que la variable N tiene el valor cero.

La entrada del procedimiento (N) es para determinar los niveles que va a tener el dibujo.

```
? PARA DIBU :N
> ESCRIBE :N
> HAZ "X ( 10 * :N ) / 2
> HAZ "Y ( 15 * :N ) / 2
> SL
> PONX :X
> PONY :Y
> PONRUMBO 0
> BL
> SI :N = 0 [ ALTO ] [ MONTE :N DIBU :N - 1 ]
> FIN
```

El otro procedimiento (MONTE) dibuja los rectángulos que corresponden a cada nivel.

```
? PARA MONTE :N
> REPITE 2 [ AV 15 GI 90 AV 10 GI 90 ]
> GI 90 AV 10 GD 90
> SI :N = 1 [ ALTO ] [ MONTE :N - 1 ]
> FIN
```

El Logo nos permite unir varias palabras en una sola.

EXPERIENCIA Y PRACTICAS EN LOGO

Para obtener el dibujo pedido ejecutamos los procedimientos:

? DIBU 12

Si cambiamos el dato de entrada obtenemos dibujos con distintos niveles, por ejemplo:

? DIBU 6

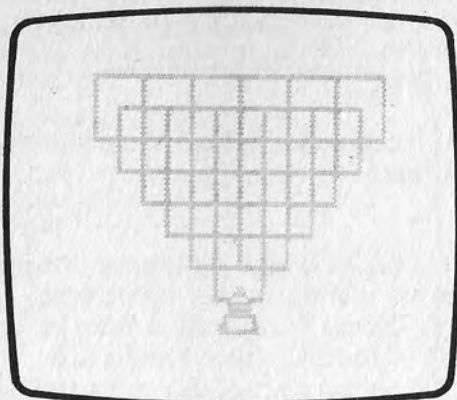


Fig. 4.

3:

Para dibujar la balanza vamos a ir definiendo procedimientos que dibujen diferentes partes de ella.

En primer lugar, definimos el que dibuja la base de la balanza:

```
? PARA BASE
> PONCL 1
> PONX -36
> AV 5 GD 90
> AV 33 GI 90
> AV 50
> REPITE 9 [ GD 18 AV 1 ]
> PONRUMBO 180
> AV 50 GI 90
> AV 33 GD 90
> AV 5 GD 90
> AV 36
> SL
> GD 90 AV 2
> BL
> RELLENA
> FIN
```

A continuación el que dibuja el soporte de los platillos:

```
PARA SOPORTE
> SL
```

```
> PONCL 3
> PONPOS [ 0 45 ]
> BL
> GI 90 AV 30
> GD 90 AV 5
> GD 90 AV 4
> GD 90 AV 2
> GD 90 AV 2
> GD 90 AV 2
> GD 90 AV 48
> GD 90 AV 2
> GD 90 AV 2
> GD 90 AV 2
> GD 90 AV 4
> GD 90 AV 5
> GD 90 AV 30
> SL
```

```
> PONRUMBO 0
> AV 2
> BL
> RELLENA
> SL
> GD 90 AV 10
> BL
> RELLENA
> SL
> RE 20
> BL
> RELLENA
> PONCL 0
> SL
> PONPOS [ 0 48 ]
> BL
> REPITE 4 [ AV 1 GD 90 ]
> FIN
```

El que dibuja los platillos es:

```
? PARA PLATILLOS
> PONCL 3
> PONRUMBO 180
> BL
> AV 5 GD 20
> AV 30
> SL
> RE 30
> BL
> GI 40 AV 30
> PONRUMBO 270
> AV 20
> PONRUMBO 135
> AV 1
```

Los números son caracteres especiales que pueden ir sin las comillas.

```

> PONRUMBO 90
> AV 18
> PONRUMBO 45
> AV 1
> FIN

```

Y por último, el que dibuja las pesas:

```

? PARA PESA
> PONRUMBO 0
> BL
> REPITE 2 [ AV 8 GD 90 AV 6 GD 90 ]
> AV 8 GD 90
> AV 2 GI 90
> AV 2 GI 90
> AV 1 GD 90
> AV 1 GD 90
> AV 4 GD 90
> AV 1 GD 90
> AV 1 GI 90
> AV 2
> FIN

```

Una vez que tenemos todas las partes, definimos el procedimiento que inicializa el proceso y agrupa a todos los que ya hemos definido:

```

? PARA BALANZA
> BP
> OT
> BL
> BASE
> SOPORTE
> SL
> PONPOS [ -25 44 ]
> PLATILLO
> SL
> PONPOS [ 25 44 ]
> PLATILLO
> SL
> PONPOS [ -28 12 ]
> PESA
> SL
> PONPOS [ 22 12 ]
> PESA
> FIN

```

Para ejecutarlo:

```

? BALANZA

```

4:

En primer lugar, vamos a definir el procedimiento el cual va a ir pidiendo los datos necesarios para resolver el problema:

```

? PARA DATOS
> BT
> PT
> ES "INTRODUCE$ MASA1
> HAZ "M1 LC
> ES :M1
> ES "INTRODUCE$ MASA2
> HAZ "M2 LC
> ES :M2
> ES "INTRODUCE$ ROZAMIENTO
> HAZ "R LC
> HAZ "R :R / 10
> ES :R
> ES INTRODUCES$ VELOCIDAD$ INICIAL
> HAZ "V :V / 10
> ES :V
> CALCULO
> FIN

```

Este procedimiento va pidiendo que se introduzca:

- La masa del primer cuerpo.
- La masa del segundo cuerpo.
- El coeficiente de rozamiento.
- La velocidad inicial.

Todos los datos no pueden ser mayores de nueve, ya que sólo se permite pulsar una tecla.

Tanto el valor del coeficiente de rozamiento como el valor de la velocidad que se introducen, son divididos por diez, para que los valores no sean excesivamente grandes. Por esto, ambos valores no podrán ser superiores a 0.9, y se obtendrá cuando se introduzca un nueve.

Una vez que tenemos todos los datos del problema, pasamos al cálculo y posteriormente a dar la solución del problema dependiendo del resultado obtenido:

```

? PARA CALCULO
> BT
> HAZ "A ( :M1 - ( :R * :M2 ) ) * 9.8 / (
: M1 + :M2 )

```

Algunas versiones del Logo disponen de órdenes de varios caracteres al mismo tiempo

EXPERIENCIA Y PRACTICAS EN LOGO

```
> SI :A = 0 [ SI : V = 0 [ RES ALTO ] [
  ES "LA$ VELOCIDAD$ ES$ SIEMPRE$
  IGUAL$ A: ES :V ALTO ] ]
> SI :A < 0 [ RES ALTO ]
> ES "LA$ ACELERACION$ ES:
> ES :A
> FIN
```

El procedimiento RES escribe una de las respuestas posibles:

```
? PARA RES
> ES "EL$ SISTEMA$ NO$ SE$ MUE-
  VE$
> FIN
```

Calculamos el valor de la aceleración y lo guardamos en la variable :A.

Una vez calculada, y dependiendo de su valor y del valor de la velocidad, obtenemos las diferentes respuestas.

Si la aceleración que se ha obtenido, no es menor ni igual a cero, se escribe el mensaje:

· LA ACELERACION ES

y a continuación el valor de ésta que está en la variable :A.

Si la aceleración es igual a cero y la velocidad también, se escribe el mensaje:

EL SISTEMA NO SE MUEVE

El mismo mensaje se escribe cuando la aceleración es menor que cero.

Si la aceleración es igual a cero, pero la velocidad no, se escribe el mensaje:

LA VELOCIDAD ES SIEMPRE IGUAL A

y a continuación el valor de la velocidad que se ha introducido que se encuentra en la variable :V.

Para ejecutar el procedimiento introducimos:

DATOS

5:

```
— ? ESCRIBE "P2,E%E
CORRECTO.
```

Se ordenará que se escriba una palabra compuesta de diferentes caracteres.

```
— ? ESCRIBE "DONDE ESTAN LAS
  LLAVES
```

INCORRECTA.

No se pueden escribir varias palabras separadas de un espacio si no se utilizan los caracteres especiales, \$ ó /.

La forma correcta de indicarlo es:

```
? ESCRIBE "DONDE$ ESTAN$ LAS$
  LLAVES
— ? PARA A
> HAZ "V "LUIS
> HAZ "W "110
> AV :V
> RE :W
> FIN
```

Este procedimiento tiene solamente una orden que no es correcta:

AV :V

A la variable :V se le asigna una palabra, por lo que es imposible utilizarla con una orden que necesite un valor numérico.

```
— ? PARA E2
> ESCRIBE PALABRA? "
> FIN
```

CORRECTO.

Si ejecutamos E2 en la pantalla aparecerá CIERTO. Estamos preguntando si la palabra vacía es una palabra.

```
— ? PARA E3
> HAZ "B LEECAR
> SI ASCII :B = 13 [ ALTO ]
> ESCRIBE PALABRA :B
> E3
> FIN
```

CORRECTO.

El procedimiento E3 acepta caracteres del teclado y los escribe en la pantalla, hasta que pulsemos la tecla RETURN.

```
— ? HAZ "C (PALABRA "FAL "SO)
? ESCRIBE :C
```

PALABRA sólo tiene dos datos, no son precisos los paréntesis, pero si los incluimos no cometemos ningún error.

Con el LOGO podemos realizar procesos en los que la tortuga no intervenga.

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS



CONTINUANDO con las series dedicadas al color, vamos a ver en este nuevo tomo cómo utilizar el color en los ordenadores IBM pc y compatibles.

El color en el IBM

Antes de empezar a explicar la utilización del color en el IBM es necesario advertir que, todo lo que sobre él se diga, será referido a aquellos ordenadores que dispongan de tarjeta de gráficos.

Una vez visto esto, diremos que, desde el BASIC, disponemos de tres modos distintos de pantalla. Cada tipo se diferencia de los otros en la capacidad de resolución gráfica que es capaz de procesar y generar. Así nos encontramos con que tenemos un modo de texto, otro de media resolución y otro de alta resolución.

Para cambiar de un modo a otro utilizaremos el comando SCREEN (que en inglés significa pantalla) seguido de un número que le dice al ordenador en qué tipo de pantalla queremos trabajar.

SCREEN 0 — Modo texto.

SCREEN 1 — Modo de media resolución.

SCREEN 2 — Modo de alta resolución.

Como todavía no hemos visto cómo dibujar con nuestro ordenador, no vamos a explicar en qué consiste la media y la alta resolución. Sólo diremos que el número de colores que podemos utilizar en cada uno de los modos de pantalla es distinto, y eso será lo que estudiaremos.

La segunda aclaración que hay que hacer es que todo lo que se diga sobre color y

gráficos está referido al GWBASIC de Microsoft Inc.

El IBM (y todos sus compatibles) tienen la posibilidad de tener 16 colores. Estos son:

- 0 — Negro
- 1 — Azul oscuro
- 2 — Verde
- 3 — Cian o Azul claro
- 4 — Rojo
- 5 — Magenta Morado
- 6 — Marrón
- 7 — Blanco
- 8 — Gris
- 9 — Azul oscuro brillante
- 10 — Verde brillante
- 11 — Cian o Azul claro brillante
- 12 — Rojo brillante
- 13 — Magenta o Morado brillante
- 14 — Amarillo
- 15 — Blanco brillante

Aunque tenemos muchos colores, estos no podrán ser utilizados todos a la vez en todos los tipos de pantalla. En la tabla que aparece a continuación se nos indica cuántos colores pueden ser utilizados en cada tipo de pantalla.

Modo de pantalla = 0 → 16 colores
Modo de pantalla = 1 → 4 colores
Modo de pantalla = 2 → 2 colores

Antes de continuar con los colores, fijémonos en la pantalla. Aunque a simple vista no se ve, está dividida en dos partes. La central es la pantalla propiamente dicha y será en ella donde podamos escribir. La parte que rodea a esta parte central es el BORDE y no podremos escribir en él, pero sí cambiarle el color.

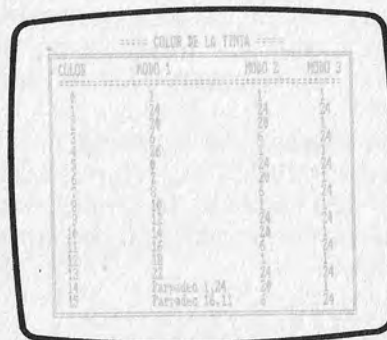
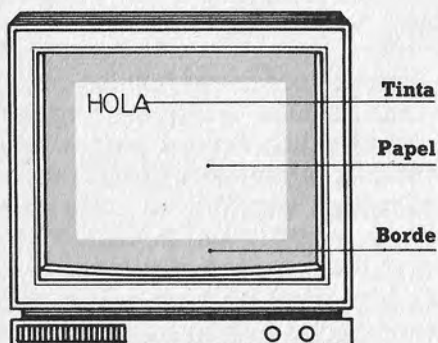
El programa número 1 que aparece a continuación, nos muestra los 16 colores de que dispone el IBM. Dichos colores nos van a aparecer en el borde de la pantalla.

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```

100 REM *****
110 REM * PROGRAMA DEMO PARA VER *
120 REM * LOS 16 COLORES DEL IBM *
130 REM *****
140 REM
150 SCREEN 0
160 DIM A$(16)
170 RESTORE
180 FOR I=1 TO 16
190   READ A$(I)
200 NEXT I
210 REM
220 REM *** LINEAS DE DATA ***
230 REM
240 DATA "Negro","Azul oscuro","Verde","Cian o Azul claro","Rojo"
250 DATA "Magenta o Morado","Marron","Blanco","Azul oscuro brillante"
260 DATA "Verde brillante","Cian o Azul claro brillante","Rojo brillante"
270 DATA "Magenta o Morado brillante","Amarillo","Blanco brillante"
280 REM
290 REM *** PROGRAMA PRINCIPAL ***
300 REM
310 FOR K=1 TO 2 STEP 0
320   FOR I=0 TO 15
330     CLS
340     PRINT "El color del borde es: ";A$(I+1)
350     COLOR ,,I
360     PRINT:PRINT:PRINT
370     PRINT "  PULSA UNA TECLA"
380     PRINT " ('F' para terminar)"
390     LET B$=INPUT$(1)
400     IF B$="f" OR B$="F" THEN GOTO 430
410   NEXT I
420 NEXT K
430 END
  
```

En este programa hemos podido ver cómo lo único que cambia de color es el borde de la pantalla. Esta es la parte donde no podemos escribir. La zona central y rectangular que no cambiaba de color es la única parte donde podemos hacerlo.



Para entender mejor cómo podemos cambiar el color de la pantalla propiamente dicha, nos tenemos que imaginar que es una hoja de papel. Como es lógico, dicha hoja puede ser de un color cualquiera y, como también es lógico, lo que escribamos en dicho papel, las letras que vamos a imprimir, también pueden ser de colores. Por ello tenemos la posibilidad de cambiar el color del papel y de la tinta de las letras separadamente. Esto lo podemos realizar con la ayuda de la sentencia

COLOR. La sintaxis de esta orden es la siguiente:

COLOR color de las letras, color del papel, color del borde.

Para entender un poco mejor esto vamos a ver el programa número 2. Con él podremos ver todas las combinaciones posibles de papel y tinta. Hay algunas que no se ven bien y hay otras en las cuales no se distingue nada. Esto ocurre cuando el color del papel y de la tinta coinciden.

Por otro lado hay que decir que, mientras el color de los caracteres puede ser cual-

quiera de los 16 de que disponemos, el color del papel sólo puede tomar valores entre 0 y 7. Si en el segundo parámetro de la sentencia COLOR ponemos un número mayor de siete, el ordenador dividirá dicho número por ocho y pondrá el color del papel del resto que nos salga en la operación. Por ello, si decimos:

COLOR 12,13

las letras aparecerán del color Rojo brillante y el papel de color Cian o Azul claro. Esto es debido a que $13/8$ es igual a 1 y nos queda 3 como resto. El color 3 es el Cian o Azul claro.

```
100 REM *****
110 REM * PROGRAMA DEMOS PARA VER *
120 REM * LAS DISTINTAS COMBINA- *
130 REM * CIONES DE TINTA Y PAPEL *
140 REM * EN EL IBM. *
150 REM *****
160 REM
170 SCREEN 0
180 DIM A$(16)
190 RESTORE
200 FOR I=1 TO 16
210   READ A$(I)
220 NEXT I
230 REM
240 REM *** LINEAS DE DATAS ***
250 REM
260 DATA "Negro", "Azul oscuro", "Verde", "Cian o Azul claro", "Rojo"
270 DATA "Magenta o Morado", "Marron", "Blanco", "Gris", "Azul oscuro brillante"
280 DATA "Verde brillante", "Cian o Azul claro brillante", "Rojo brillante"
290 DATA "Magenta o Morado brillante", "Amarillo", "Blanco brillante"
300 REM
310 REM *** PROGRAMA PRINCIPAL ***
320 REM
330 CLS
340 FOR K=1 TO 2 STEP 0
350   FOR J=0 TO 7
360     FOR I=0 TO 15
370       COLOR I,J
380       PRINT "Esta frase esta escrita en ";A$(I+1); " sobre ";A$(J+1)
390       FOR Z=1 TO 300
400         NEXT Z
410       NEXT I
420     NEXT J
430   NEXT K
440 END
```

Aunque no lo hemos dicho, es lógico pensar que cada letra puede ser de un color y que el fondo sobre el que se imprime cada letra también puede ser de cualquier color in-

dependientemente del color de la letra o del espacio siguiente. El programa número 3 nos muestra un ejemplo de ello.

```
100 REM *****
110 REM * PROGRAMA DEMO PARA VER *
```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```
120 REM * QUE CADA CARACTER DEL *
130 REM * IBM PUEDE TENER SU PRO *
140 REM * PIO COLOR DE PAPEL Y *
150 REM * DE TINTA. *
160 REM *****
170 REM
180 SCREEN 0
190 COLOR 7,0,0
200 CLS
210 PRINT "Introduce el mensaje que quieres imprimir."
220 PRINT
230 LINE INPUT "=="> ";A$
240 PRINT
250 PRINT STRING$(80,"*")
260 PRINT
270 FOR J=0 TO 7
280   LET T=0:REM <-- COLOR DE LA TINTA
290   FOR I=1 TO LEN(A$)
300     IF J=T THEN LET T=T+1
310     IF T=16 THEN LET T=0:GOTO 300
320     COLOR T,J
330     PRINT MID$(A$,I,1);
340     LET T=T+1
350   NEXT I
360   PRINT
370   PRINT
380 NEXT J
390 COLOR 7,0
400 PRINT STRING$(80,"*")
410 LOCATE 1,1
420 END
```

Al ejecutar el programa podemos apreciar que nunca se pone la tinta del mismo color que el papel. Esto se ha solucionado introduciendo la línea 300 que controla que ambos colores no sean iguales. Aun así, hay letras que no se pueden ver del todo bien debido a que tienen un color muy parecido al de su fondo.

Por otro lado, lo mismo que hemos hecho con la tinta, cambiándola de color cada vez que escribimos una letra, lo podemos hacer con el papel.

Para terminar de ver el color en el modo 0 de pantalla (SCREEN 0) sólo nos queda ver el parpadeo. El IBM nos da la posibilidad de escribir mensajes que parpadeen. Esto lo podemos conseguir con sólo poner, en la instrucción COLOR, un número mayor de 15.

Por ejemplo, si escribimos:

COLOR 16

todo lo que escribamos a continuación aparecerá en negro parpadeante. Según esto, se puede ver que si el color 16 es negro parpadeante, el 17 será azul oscuro parpadeante, el 18 verde parpadeante y así sucesivamente hasta el color 31. Esto sólo es válido para la tinta, ya que no se puede hacer que el papel parpadee.

Dicho parpadeo consiste en imprimir el texto que queramos primero en color n-16 (donde n es el número que hemos puesto en la instrucción COLOR) y después borrarlo, o lo que es igual, imprimirlo del mismo color que el papel. El programa número 4 nos muestra más claramente todo esto.

```
100 REM *****
110 REM * PROGRAMA DEMO PARA VER *
120 REM * COMO SON LOS COLORES *
130 REM * PARPADEANTES EN EL IBM *
140 REM *****
150 REM
```

```

160 SCREEN 0
170 COLOR 7,0
180 CLS
190 PRINT "Dime que quieres que escriba."
200 PRINT
210 LINE INPUT "=> ";A$
220 LET N=LEN(A$)
230 LET A$=LEFT$(STRING$(80," "), (80-N)\2)+A$+LEFT$(STRING$(80," "), (80-N)\2)
240 IF LEN(A$)<>80 THEN LET A$=A$+" "
250 PRINT
260 INPUT "Con que tinta";T
270 PRINT
280 IF T<0 OR T>15 THEN PRINT "Dame un numero entre 0 y 15.":GOTO 250
290 INPUT "Con que papel? ";P
300 PRINT
310 IF P<0 OR P>7 THEN PRINT "Dame un numero entre 0 y 7.":GOTO 270
320 PRINT STRING$(80,"*")
330 REM
340 REM *** ESTA ES LA INSTRUCCION QUE GENERA EL PARPADEO ***
350 REM
360 COLOR T+16,P
370 PRINT STRING$(80," ")
380 PRINT A$
390 PRINT STRING$(80," ")
400 COLOR 7,0
410 PRINT STRING$(80,"*")
420 PRINT:PRINT
430 END

```

Con esto hemos terminado todo lo referente al color en el modo 0 de pantalla. A continuación vamos a ver cómo funciona y cómo tratar el color en los otros dos modos.

En el modo 1 de pantalla (SCREEN 1) no podemos tener más de cuatro colores en pantalla. Uno de ellos como color del papel y los otros tres como color de la tinta. Por otro lado,

no podemos elegir qué tres colores queremos utilizar, sino que tenemos definidas dos paletas, con distintos colores cada una de ellas, de las que no nos podemos salir. Por el contrario, siempre podemos variar el color del papel, pero se nos cambiará todo el papel de la pantalla de golpe. Para entender un poco esto vamos a ver el programa número 5.

```

100 REM *****
110 REM *   PROGRAMA DEMO PARA VER LOS DISTINTOS *
120 REM * COLORES QUE PODEMOS UTILIZAR EN EL IBM *
130 REM * CUANDO ESTA EN EL MODO 1 DE PANTALLA. *
140 REM *****
150 REM
160 DIM A$(16)
170 RESTORE
180 FOR I=1 TO 16
190   READ A$(I)
200 NEXT I
210 REM
220 REM *** LINEAS DE DATA ***
230 REM
240 DATA "Negro", "Azul oscuro", "Verde", "Cian o Azul claro", "Rojo"
250 DATA "Magenta o Morado", "Marron", "Blanco", "Gris", "Azul oscuro brillante"
260 DATA "Verde brillante", "Cian o Azul claro brillante", "Rojo brillante"
270 DATA "Magenta o Morado brillante", "Amarillo", "Blanco brillante"
280 REM
290 REM *** PROGRAMA PRINCIPAL ***
300 REM
310 SCREEN 1

```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```
320 CLS
330 COLOR 0,0
340 PRINT "PALETA NUMERO 0"
350 PRINT
360 FOR I=1 TO 3
370     COLOR ,,,,I
380     FOR J=1 TO 5
390         PRINT STRING$(40,219)
400     NEXT J
410 NEXT I
420 PRINT
430 FOR K=0 TO 1
440     FOR I=1 TO 16
450         LOCATE 22,1
460         PRINT "EL PAPEL ESTA DE COLOR ...."
470         LOCATE 23,1
480         PRINT SPACE$(40)
490         LOCATE 23,1
500         PRINT A$(I)
510         COLOR 16*K+I-1
520         FOR J=1 TO 500
530             NEXT J
540     NEXT I
550     LOCATE 1,1
560     PRINT "PALETA NUMERO 1"
570 NEXT K
580 COLOR 0,0
590 END
```

En este programa vemos cómo los colores del papel van cambiando uno a uno. También podemos apreciar las dos paletas de que disponemos. Estas dos paletas están numeradas como 0 y 1 y los colores que tienen son los siguientes:

PALETA 0 = Verde, Rojo, Marrón
PALETA 1 = Cian, Magenta, Blanco

La forma de cambiar el color del papel es mediante el primer argumento de la instrucción color.

.. COLOR n nos cambia el color del papel y lo pone de color 'n'.

Para cambiar el color de la paleta tenemos dos maneras distintas. La primera es poner, en el primer argumento de la sentencia COLOR un número mayor de 15. Al poner este número el papel de la pantalla nos cambiará y la paleta que veremos será la número 1. Por ello, el color 16 es negro, pero con paleta de colores 1. El color 20 es rojo con paleta de colores 1.

Para volver a utilizar la paleta número 0 sólo tenemos que poner un número entre 0 y 15. Con ello cambiaremos el color del papel y nos encontraremos en la paleta número 0.

Según esto, los colores que van desde el 0 al 15, me cambia el color del papel y los

colores de tinta que podemos utilizar son los de la paleta número 0. Si ponemos un número entre el 16 y 31 en la sentencia color, el color de la pantalla se pondrá como n-16 y la paleta de la tinta de la que disponemos será la número 1.

La otra forma de cambiar el color de la paleta es poner un 0 o un 1 en el segundo argumento de la sentencia COLOR dependiendo de la paleta elegida.

COLOR, n

Una vez que hemos elegido una de las paletas para cambiar el color de las letras que vamos a escribir utilizamos el quinto argumento de la sentencia color.

COLOR ,,,,n

donde 'n' es un número entre 1 y 3. El color resultante dependerá de la paleta que tengamos en dicho momento. La siguiente tabla nos muestra los números de cada color según la paleta.

COLOR	PALETA 0	PALETA 1
1	VERDE	CIAN
2	ROJO	MAGENTA
3	MARRON	BLANCO

Visto todo esto, la situación de la sentencia COLOR nos queda como sigue:

COLOR color de papel, paleta, no usado, no usado, color de tinta.

Por otro lado, este modo de pantalla nos permite que, al imprimir caracteres en la pantalla, éstos se impriman de forma normal o realizando la función XOR con contenido de la pantalla en el momento de imprimir. Para ello

sólo tenemos que poner en el quinto argumento de la sentencia COLOR un número entre 5 y 7. Este número le dice al ordenador que lo siguiente que escriba tiene que hacerlo realizando la función XOR. El color número 5 es color y el número 1, pero con función XOR. El número 6 es como el número 2 y el número 7 color el número 3.

Para entender mejor esto vamos a ver el programa número 6.

```
100 REM *****
110 REM * PROGRAMA DEMO PARA VER EL USO *
120 REM * DEL XOR EN EL IBM *
130 REM *****
140 SCREEN 1
150 CLS
160 PRINT "QUE QUIERES QUE ESCRIBA"
170 PRINT
180 INPUT "==> ";A$
190 PRINT
200 FOR I=1 TO 3
210   COLOR ,,,I
220   PRINT "ESCRIBO TU MENSAJE CON COLOR";I
230   PRINT
240   PRINT A$
250 PRINT
260 NEXT I
270 PRINT
280 PRINT "PULSA UNA TECLA"
290 LET B$=INPUT$(1)
300 CLS
310 PRINT "AHORA VOY A ESCRIBIRLO Y A REESCRIBIRLO EN LA MISMA POSICION"
320 PRINT
330 FOR I=5 TO 7
340   LOCATE 15,1
350   COLOR ,,,I
360   PRINT A$
370   FOR K=1 TO 500
380     NEXT K
390   LOCATE 15,1
400   PRINT A$
410   FOR K=1 TO 500
420     NEXT K
430 NEXT I
440 PRINT "PULSA UNA TECLA"
450 LET B$=INPUT$(1)
460 CLS
470 PRINT "AHORA VOY A ESCRIBIR TU MENSAJE Y VOY A SOBREIMPRIMIR UNA FILA DE 'A'"
480 FOR I=5 TO 7
490   COLOR ,,,I
500   LOCATE 15+(I-5)*2,1
510   PRINT A$
520   LOCATE 15+(I-5)*2,1
530   FOR J=1 TO LEN(A$)
540     PRINT "A";
550   NEXT J
560 NEXT I
570 COLOR 0,0,,,1
580 END
```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

Con esto terminamos todo lo concerniente al modo 1 de pantalla. En el modo 2 de pantalla (SCREEN 2) trabajamos siempre con 80 columnas. En este caso no podemos variar ni el color del papel ni el de la tinta. Cuando nos encontramos en este modo siempre tenemos que escribir con letras blancas sobre fondo negro.

En este modo, aunque podemos utilizar la sentencia COLOR, ésta no tiene ningún efecto. Sólo su tercer argumento sirve para algo. Este nos permite escribir realizando la operación XOR con el contenido de la pantalla. Su función es la misma que la del quinto argumento de dicha sentencia cuando nos encontramos en el modo 1 de pantalla.

TRUCOS Y RUTINAS BASICAS



CONTINUANDO con lo que llevamos viendo desde hace algunos fascículos, vamos a ver a continuación un programa que nos permitirá definir nuestros propios caracteres en el AMSTRAD. El programa nos permitirá uti-

lizar cualquiera de los caracteres que ya tiene nuestro ordenador definidos para realizar nuestros propios caracteres. Con ello podemos tener un juego de letras distinto del que usamos normalmente, cambiar los números, y, en general, realizar todo aquello que se nos ocurra.

Como se puede ver en la figura número 1, la pantalla está dividida en tres partes.

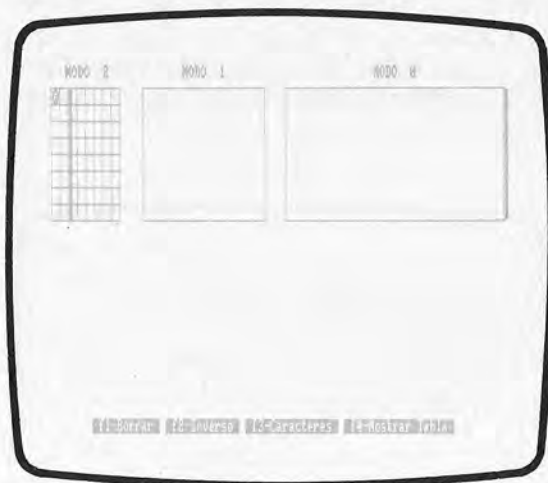


Fig. 1.

Por otro lado, la parte superior también puede ser dividida en tres partes distintas. Cada una de estas partes nos va a mostrar cómo se vería el carácter que estamos definiendo en cada uno de los tres modos de pan-

talla de que disponemos. Según vayamos dibujando el nuevo carácter, éste se irá dibujando en los tres cuadros que hay en la parte superior de la pantalla y que encima tienen el rótulo que nos dice en qué modo se vería como lo estamos viendo.

A la hora de definir nuestro carácter sólo podemos definirlo en el cuadro que está nombrado como MODE 2. Como se puede apreciar este cuadrado es el único que tiene cuadrícula. Esta nos servirá para poder definir mejor nuestro carácter.

Para saber qué punto del carácter es el que podemos pintar o borrar utilizaremos un carácter. Este es el que aparece en la esquina superior izquierda de la cuadrícula.

Para mover dicho cursor utilizaremos las teclas del cursor. Una vez que el cursor esté colocado sobre uno de los cuadrados de la cuadrícula que nosotros queramos pintar o borrar, basta con pulsar la barra espaciadora. Si el punto estaba iluminado entonces lo borraremos (lo apagaremos). Si estaba apagado entonces le pondremos color (lo encenderemos). Cada vez que hagamos esto, dicho punto aparecerá (o se borra, según el caso) en los cuadros que lleven como rótulo MODE 0 y MODE 1.

En la parte inferior de la pantalla nos encontramos con un minimenú que nos indica las posibles opciones de que dispone el programa. Estas son:

F1 — BORRAR

Pulsando esta opción el carácter que en este momento estamos editando se borra de la parrilla y se perderá su definición. Este comando es útil cuando hayamos terminado la definición de un carácter. Cuando pulsamos esta opción la pantalla nos aparece como si hubiésemos hecho RUN de nuevo en el

TRUCOS Y Rutinas BASICAS

programa. Su aspecto es el de la figura 1.

F2 — INVERSO

Al pulsar esta opción hacemos que todos los puntos que están iluminados se apaguen y que todos los puntos que están apagados se iluminen. Esta opción es muy interesante si queremos tener caracteres cuya forma venga dada por los contornos de dicho carácter. Un ejemplo de cómo se encuentra la pantalla tras pulsar esta opción lo podemos ver en la figura 2. En ella podemos ver el carácter ET (o AND, según los ingleses) en modo inverso.

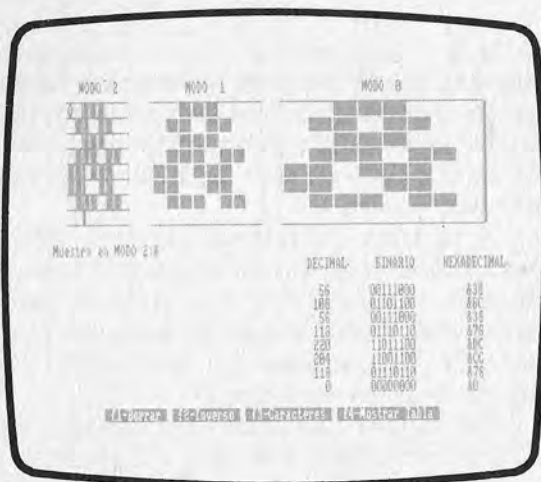


Fig. 2.

F3 — CARACTERES

Con esta opción podemos hacer que un carácter de los que tiene definidos el ordenador aparezca en nuestra parrilla de diseño. Al elegir esta opción, en la parte central de la pantalla nos aparecen unos cuantos caracteres. Moviéndonos con las teclas del cursor derecha y cursor izquierda podremos elegir el carácter a modificar. Una vez elegido basta con pulsar la tecla de retorno de carro, para que dicho carácter nos aparezca en la parrilla. En la figura 3 podemos ver cómo hemos colocado el cursor (signo de elevado a...) debajo del signo ET y hemos pulsado RETURN. También podemos ver cómo dicho carácter ya está en la parrilla y en las ventanas que nos muestran cómo se ve dicho carácter en los distintos modos.

F4 — MOSTRAR TABLA

Si elegimos esta opción, nos aparecerá en la parte central derecha de la pantalla la

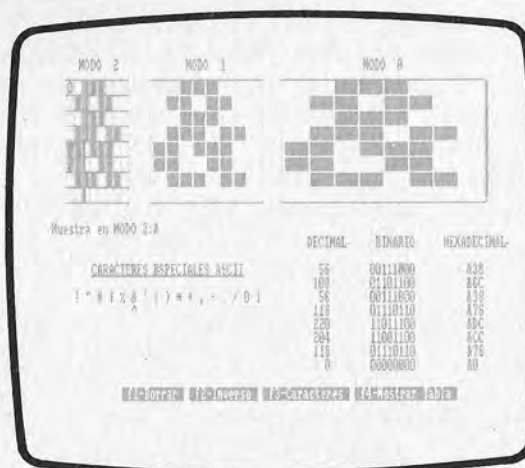


Fig. 3.

codificación numérica del carácter que estamos o acabamos de definir. Dicha configuración nos aparece en decimal, binario y hexadecimal en tres columnas distintas. Una vez que hayamos terminado de definir nuestro carácter, tenemos que pulsar esta opción para apuntar los números que aparecen. Con estos números podremos, ya dentro de nuestro propio programa, definir cualquiera de los caracteres de que dispone nuestro ordenador, como el carácter que acabamos de dibujar. Un ejemplo de cómo aparece dicha tabla en pantalla lo podemos ver en la figura 4. En ella vemos qué números son los que definen el carácter ET. Si el lector se fija en la representación en binario, podrá apreciar que es igual a la de la parrilla y que, por tanto, podemos distinguir perfectamente la forma de dicho carácter.

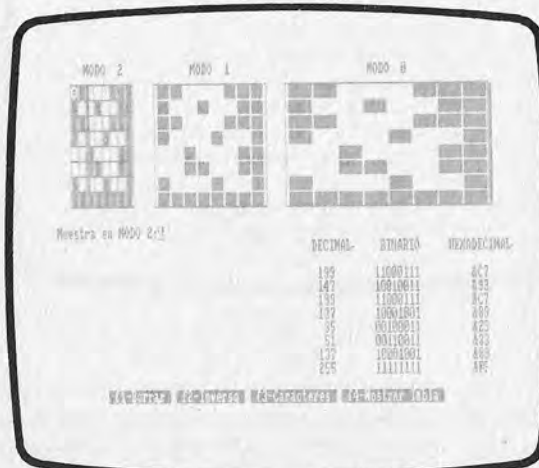


Fig. 4.

```

10 REM *****
20 REM ***      SYMBOL-HELP      ***
30 REM *** Un programa realizado ***
40 REM ***      Por      ***
50 REM *** Carlos A. Maria Morin ***
60 REM ***      ***
70 REM ***      (C) Ediciones      ***
80 REM *** Siglo Cultural 1987 ***
90 REM *****
100 REM
110 REM *****
120 REM **** INICIALIZACION Y PRESENTACION ****
130 REM *****
140 REM
150 MODE 0
160 LOCATE 1,13
170 PRINT"S Y M B O L  H E L P"
180 FOR ret=1 TO 3000
190 NEXT
200 CLS
210 MODE 2
220 mov=19
230 is=CHR$(24)
240 SYMBOL AFTER 254
250 WINDOW#1,46,80,14,23:CLS#1
260 WINDOW#2,1,45,16,19:CLS#2
270 SYMBOL 255,60,126,195,219,195,219,126,60
280 GOSUB 2020
290 LOCATE 6,1
300 PRINT"MODO 2          MODO 1          MODO 0"
310 ORIGIN 20,230
320 MOVE 120,0:DRAW 278,0
330 DRAW 278,145:DRAW 120,145
340 DRAW 120,0:MOVE 305,0
350 DRAW 590,0:DRAW 590,145
360 DRAW 305,145:DRAW 305,0
370 FOR x=1 TO 96 STEP 11
380 PLOT x,0
390 DRAW x,142
400 NEXT
410 MOVE 0,0
420 FOR y=1 TO 149 STEP 17
430 IF (y MOD 2)=0 THEN g=y+1:y=g:g=0
440 PLOT 1,y
450 DRAW 89,y
460 NEXT
470 x%=3
480 y%=142
490 REM
500 REM *****
510 REM ***** MOVIMIENTO DEL CURSOR *****
520 REM *****
530 REM
540 PRINT CHR$(23);CHR$(1);
550 GOSUB 750
560 FOR z=1 TO 100
570 NEXT
580 mov$=INKEY$
590 IF INKEY(8)=0 THEN GOSUB 750:x%=MAX(14,x%)-11:GOSUB 750
600 IF INKEY(1)=0 THEN GOSUB 750:x%=MIN(69,x%)+11:GOSUB 750
610 IF INKEY(0)=0 THEN GOSUB 750:y%=MIN(124,y%)+18:GOSUB 750
620 IF INKEY(2)=0 THEN GOSUB 750:y%=MAX(35,y%)-18:GOSUB 750
630 curx=(7-INT(x%/11)+1):cury=(9-INT(y%/16))
640 IF INKEY(47)=0 THEN GOSUB 700:CLS#1
650 IF mov$=CHR$(49) THEN GOSUB 1600

```

TRUCOS Y RUTINAS BASICAS

```

660 IF mov$=CHR$(50) THEN GOSUB 1350
670 IF mov$=CHR$(52) THEN GOSUB 1170
680 IF mov$=CHR$(51) THEN GOSUB 730:GOSUB 1600:GOSUB 2110
690 GOTO 580
700 GOSUB 860
710 GOSUB 1010
720 RETURN
730 LOCATE#2,8,1:PRINT#2,"ESPERE UN MOMENTO POR FAVOR.":RETURN
740 PRINT CHR$(23);CHR$(0);
750 TAG
760 FRAME
770 MOVE x%,y%
780 PRINT CHR$(255);
790 RETURN
800 GOSUB 750
810 REM
820 REM *****
830 REM ***** RELLENAR Y BORRAR PIXEL *****
840 REM *****
850 REM
860 TAGOFF
870 resul(cury)=0
880 posicion(curx,cury)=NOT(posicion(curx,cury))
890 FOR fila=1 TO 8
900 IF posicion(fila,cury)=-1 THEN resul(cury)=resul(cury)+(2^(fila-1))
910 NEXT
920 TAG
930 MOVE x%,y%
940 PRINT CHR$(143);
950 MOVE (x%*1.73)+120,y%
960 PRINT CHR$(143)+CHR$(143);
970 MOVE (x%*3.19)+300,y%
980 PRINT CHR$(143)+CHR$(143)+CHR$(143)+CHR$(143);
990 TAGOFF
1000 RETURN
1010 TAGOFF
1020 REM
1030 REM *****
1040 REM * DEFINICION E IMPRESION DEL CARACTER *
1050 REM *****
1060 REM
1070 SYMBOL 254,resul(1),resul(2),resul(3),resul(4),resul(5),resul(6),resul(7),r
    esul(8)
1080 LOCATE 1,13
1090 PRINT "Muestra en MODO 2: ";CHR$(254);
1100 TAG
1110 RETURN
1120 REM
1130 REM *****
1140 REM ***** IMPRESION DE LA TABLA *****
1150 REM *****
1160 REM
1170 TAGOFF
1180 CLS#1
1190 PRINT#1,"DECIMAL      BINARIO      HEXADECIMAL"
1200 PRINT#1," ";USING "###";resul(1);:PRINT#1,"      ";BIN$(resul(1),8);"
    &" +HEX$(resul(1))
1210 PRINT#1," ";USING "###";resul(2);:PRINT#1,"      ";BIN$(resul(2),8);"
    &" +HEX$(resul(2))
1220 PRINT#1," ";USING "###";resul(3);:PRINT#1,"      ";BIN$(resul(3),8);"
    &" +HEX$(resul(3))
1230 PRINT#1," ";USING "###";resul(4);:PRINT#1,"      ";BIN$(resul(4),8);"
    &" +HEX$(resul(4))
1240 PRINT#1," ";USING "###";resul(5);:PRINT#1,"      ";BIN$(resul(5),8);"
    &" +HEX$(resul(5))

```

```

1250 PRINT#1," ";USING "###";resul(6);:PRINT#1,"      ";BIN$(resul(6),8);"
      &" +HEX$(resul(6))
1260 PRINT#1," ";USING "###";resul(7);:PRINT#1,"      ";BIN$(resul(7),8);"
      &" +HEX$(resul(7))
1270 PRINT#1," ";USING "###";resul(8);:PRINT#1,"      ";BIN$(resul(8),8);"
      &" +HEX$(resul(8))
1280 TAG
1290 RETURN
1300 REM
1310 REM *****
1320 REM ***** INVIERTE EL PIXEL *****
1330 REM *****
1340 REM
1350 cury2=cury
1360 curx2=curx
1370 x2%=x%
1380 y2%=y%
1390 y%=160
1400 FOR cury=1 TO 8
1410 y%=MAX(0,y%)-18
1420 x%=-8
1430 FOR curx=1 TO 8
1440 x%=MIN(69,x%)+11
1450 GOSUB 870
1460 GOSUB 1010
1470 NEXT
1480 NEXT
1490 cury=cury2
1500 curx=curx2
1510 x%=x2%
1520 y%=y2%
1530 GOSUB 1170
1540 RETURN
1550 REM
1560 REM *****
1570 REM ***** BORRA TODOS LOS PIXELS *****
1580 REM *****
1590 REM
1600 cury2=cury
1610 curx2=curx
1620 x2%=x%
1630 y2%=y%
1640 y%=160
1650 FOR cury=1 TO 8
1660 y%=MAX(0,y%)-18
1670 x%=-8
1680 FOR curx=8 TO 1 STEP -1
1690 x%=MIN(69,x%)+11
1700 IF posicion(curx,cury)=-1 THEN GOSUB 1880
1710 resul(cury)=0
1720 posicion(curx,cury)=0
1730 GOSUB 1010
1740 TAGOFF
1750 NEXT
1760 NEXT
1770 cury=cury2
1780 curx=curx2
1790 x%=x2%
1800 y%=y2%
1810 GOSUB 1170
1820 RETURN
1830 REM
1840 REM *****
1850 REM ** GESTIONA PIXEL EN TODOS LOS MODOS **
1860 REM *****
1870 REM
1880 TAG

```

TRUCOS Y RUTINAS BASICAS

```

1890 MOVE x%,y%
1900 PRINT CHR$(143);
1910 MOVE (x%*1.73)+120,y%
1920 PRINT CHR$(143)+CHR$(143);
1930 MOVE (x%*3.19)+300,y%
1940 PRINT CHR$(143)+CHR$(143)+CHR$(143)+CHR$(143);
1950 TAGOFF
1960 RETURN
1970 REM
1980 REM *****
1990 REM ***** IMPRIME MENU *****
2000 REM *****
2010 REM
2020 LOCATE 10,25
2030 PRINT i$ " f1-Borrar "i$ " "i$ " f2-Inverso "i$ " ";
2040 PRINT i$ " f3-Caracteres "i$ " "i$ " f4-Mostrar Tabla "i$
2050 RETURN
2060 REM
2070 REM *****
2080 REM ***** COPIA Y GESTIONA CARACTERES *****
2090 REM *****
2100 REM
2110 TAGOFF
2120 x2%=x%
2130 y2%=y%
2140 CLS#2
2150 PRINT#2,CHR$(22);CHR$(1);
2160 LOCATE#2,8,1
2170 PRINT#2,"CARACTERES ESPECIALES ASCII"
2180 LOCATE#2,8,1
2190 PRINT#2,"_____";
2200 PRINT#2,CHR$(22);CHR$(0);
2210 IF INKEY(8)=0 THEN mov=mov-2
2220 IF INKEY(1)=0 THEN mov=mov+2
2230 IF INKEY(18)=0 THEN 2350
2240 IF mov<3 THEN mov=3:xx=xx-1:hh=hh-17 ELSE IF mov>35 THEN mov=35:xx=xx+1:hh=
hh+17
2250 IF xx<20 THEN xx=5 ELSE IF xx>40 THEN xx=40
2260 IF hh<32 THEN hh=32:f$=i$ ELSE IF hh>236 THEN hh=236:f$=i$
2270 LOCATE#2,xx,3
2280 FOR cc=1 TO 17
2290 PRINT#2,CHR$(hh+cc)+" ";
2300 NEXT
2310 LOCATE#2,mov,4
2320 PRINT#2," "+f$+CHR$(196)+f$+" "
2330 f$=""
2340 GOTO 2210
2350 y%=160
2360 FOR cary=8 TO 1 STEP -1
2370 y%=MAX(0,y%)-18
2380 x%=-8
2390 FOR carx=1 TO 8
2400 x%=MIN(69,x%)+11
2410 IF TEST((mov*8)+carx-13,((cary*2)-120)) THEN GOSUB 2500
2420 NEXT
2430 NEXT
2440 x%=x2%
2450 y%=y2%
2460 GOSUB 1010
2470 GOSUB 1170
2480 CLS#2
2490 RETURN
2500 posicion(9-carx,9-cary)=-1
2510 resul(9-cary)=resul(9-cary)+(2^(8-carx))
2520 GOSUB 1880
2530 RETURN

```

Una vez que hemos definido el carácter, o caracteres, que queríamos y apuntado los números que definen dicho carácter, lo único que tenemos que hacer para definir en nuestro programa un carácter con dicha forma, es utilizar la función SYMBOL. La sintaxis de esta instrucción es la siguiente:

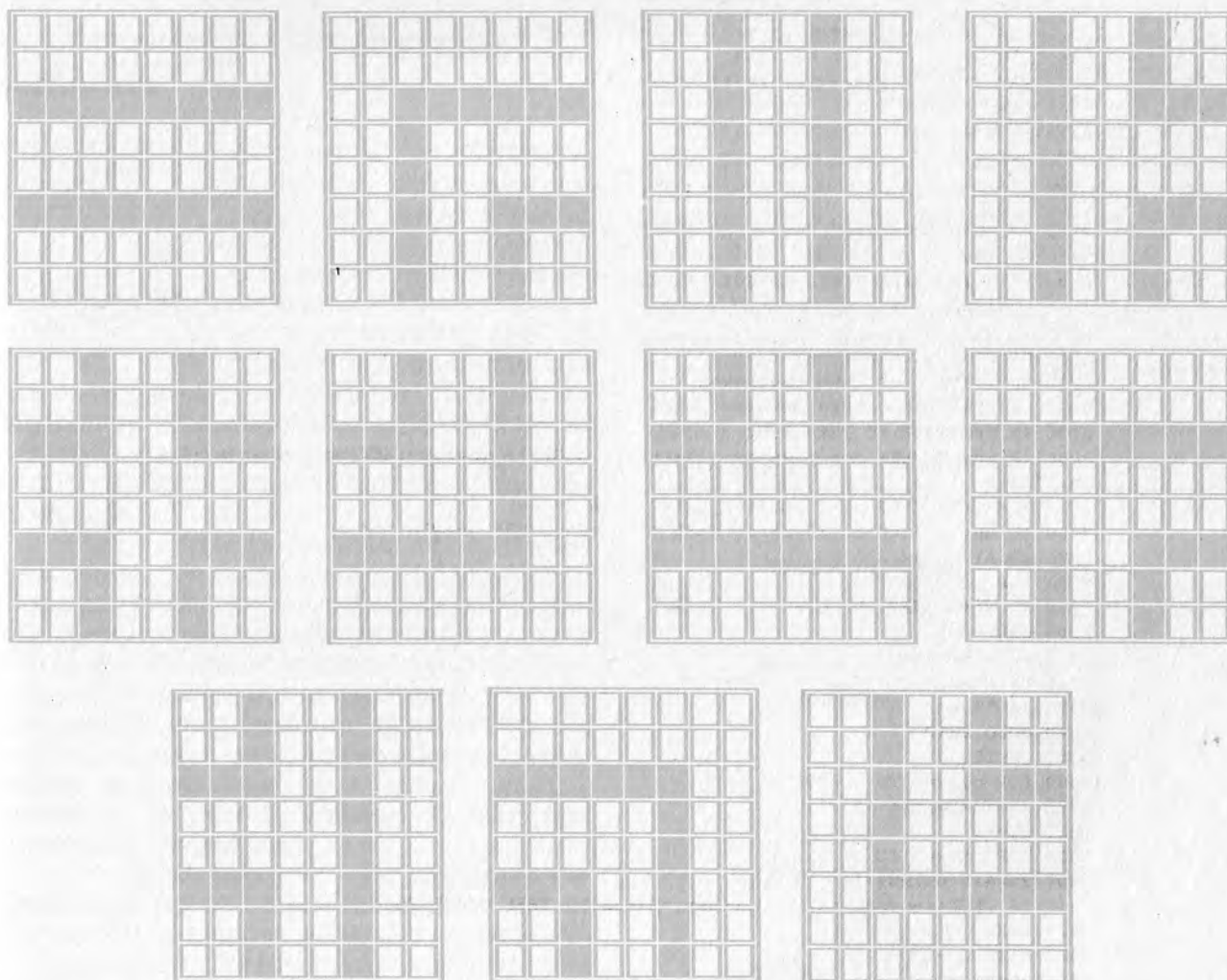
SYMBOL carácter, N1, N2, N3, N4, N5, N6, N7, N8

donde:

'carácter' es el código ASCII del carácter que queremos tenga la forma que hemos definido.

'N1, N2, ..., N8' son los ocho números que nos proporciona el programa SYMBOL-HELP. Estos números pueden estar en decimal, en binario o en hexadecimal y su orden en la instrucción es el orden en que aparecieron en la pantalla si los hubiésemos leído de arriba a abajo.

El uso del programa es muy sencillo mientras que su utilidad es evidente. Por ello, se insta al lector a que defina sus propios caracteres para aprender su manejo y entender del todo lo que aquí hemos explicado. A continuación aparecen algunos caracteres que el lector puede introducir.



■ Nota para los usuarios del Amstrad CPC464

El programa ha sido realizado en un ordenador AMSTRAD 6128. Por ello, algunos de los comandos que aparecen en el programa

nos darán 'Sintaxis error' si intentamos ejecutarlos. Por otra parte este ordenador carece de teclas de función.

Por todo ello, vamos a ver algunas modificaciones que hay que hacerle al programa para que funcione en dicho ordenador. Entre

TRUCOS Y RUTINAS BASICAS

ellas, vamos a cambiar las líneas que se encargan de comprobar si hemos pulsado algunas de las teclas de función. Cambiaremos las teclas que nos van a permitir borrar, invertir y ver los caracteres, así como ver la tabla de números. Las teclas que tendrán que pulsar los usuarios del CPC464 son las siguientes:

B — Para BORRAR la parrilla.

I — Para INVERTIR la parrilla.

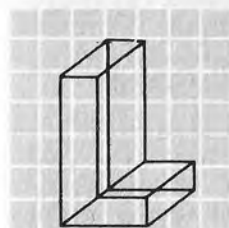
C — Para elegir el CARACTER a modificar.

T — Para ver la TABLA de los números.

Dichas modificaciones son:

```
650 IF MOV$ = "b" OR MOV$ = "B" THEN
GOSUB 1600
660 IF MOV$ = "i" OR MOV$ = "I" THEN
GOSUB 1350
670 IF MOV$ = "t" OR MOV$ = "T" THEN
GOSUB 1170
680 IF MOV$ = "c" OR MOV$ = "C" THEN
GOSUB 730; GOSUB 1600; GOSUB 2110
760 REM
2030 PRINT I$ B —Borrar "I$" "I$" I —In-
verso "I$" ";
2040 PRINT I$ C —Caracteres "I$" "I$"
T—Mostrar Tabla "I$"
```

■ Comunicación entre dispositivos periféricos



A conexión al ordenador personal de muchos periféricos ya existentes se puede englobar dentro de las conexiones entre máquinas inteligentes. El problema de la interconexión, por su complejidad, ha hecho que

sea en este área donde más han trabajado los organismos internacionales para facilitar el desarrollo de la tecnología mediante la normalización de las conexiones entre máquinas de diferentes fabricantes.

Los periféricos podemos considerarlos de dos tipos: los que se encuentran muy próximos a la máquina central y los que se encuentran alejados. Los primeros pueden compartir con la unidad central todos los circuitos internos, entre ellos los buses de datos y direcciones, además de la fuente de alimentación. Los segundos suelen estar alimentados por separado y se comunican con la unidad central a través de un número limitado de hilos que transportan las señales.

Podemos distinguir dos formas de conexión básicas: en serie y en paralelo. Vamos a presentar los rasgos generales de cada forma de conexión, las normas que son aplicables a estos casos y después se describirán proyectos de aplicación para conectar algunos dispositivos a nuestras máquinas.

■ Conexiones en serie

Una forma muy extendida de conexión de periféricos es mediante circuitos en serie.

La conexión en serie significa que los datos se presentan de manera secuencial en el canal de comunicación. Para velocidades de transferencia no muy altas permite reducir el número de hilos a emplear si se utiliza información codificada en forma serie. Hay un número muy elevado de dispositivos que pueden conectarse de forma serie asíncrona a cualquier ordenador personal y que con pequeñas adaptaciones de programación pueden integrarse dentro de cualquier aplicación. Es por esta razón muy interesante disponer de conexión serie asíncrona en todo ordenador personal y de hecho la mayoría de los que actualmente se comercializan disponen de esta conexión de forma estándar o como opción económica.

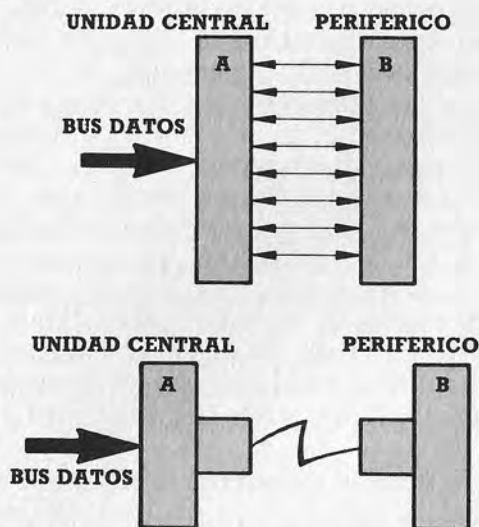


Fig. 1. Conexiones en paralelo y en serie.

Empezaremos por describir los rasgos generales de las interconexiones en serie más usuales y se presentarán aplicaciones de co-

EL TALLER DEL HARDWARE

nexión entre dos ordenadores personales, a través de un puerto serie para diferentes tipos de aplicación: Utilización de terminal de entrada de datos, transferencia de ficheros, comunicaciones, un ordenador como periférico del otro, etc.

Es conveniente conocer los métodos recomendados de interconexión y que se han plasmado como normas de los organismos internacionales, por lo que daremos algunas explicaciones oportunas.

■ Normas de interconexión

Para hacer posible la interconexión de periféricos, independientemente del fabricante del ordenador, se han extendido como normas de hecho las recomendaciones dadas por organismos internacionales (EIA = Electric Industry Association, UIT = Unión Internacional de Telecomunicaciones, CCITT = Comité Consultivo para el Intercambio Telefónico y Telegráfico, ISO = International Standard Organization). Estas recomendaciones y normas especifican las características mecánicas, eléctricas y de procedimiento a emplear para hacer compatibles los equipos. Mediante la divulgación del empleo de las normas se ha originado la necesidad de que numerosos fabricantes ofrezcan soluciones competitivas para cada parte de la conexión: circuitos integrados para la interfaz y para la manipulación de la señal física, conectores, cables y hasta programas que directamente soportan el nivel correspondiente de la conexión.

La conexión serie implica el encadenamiento secuencial de elementos de señal descriptivos de la información a enviar. La información estará inicialmente codificada, generalmente en código también normalizado como el ASCII o el EBCDIC. La transmisión en serie envía los elementos de cada código secuencialmente en un orden determinado.

La sincronización en la transmisión, para identificar cada carácter enviado puede realizarse con dos métodos: asíncrono y síncrono.

El **método asíncrono** utiliza como identificador de comienzo de carácter un cambio de nivel y como finalización del carácter otro cambio de nivel.

Los símbolos utilizados para indicación de principio y fin tienen la duración igual a la de los elementos de información, identificando la existencia de un carácter por el instante

de aparición, a partir del reposo. La línea en reposo se mantiene a un nivel correspondiente al estado 1. El comienzo de carácter se indica por un cambio a nivel 0, enviándose a continuación los símbolos del carácter correspondiente. Después del último bit se envía el símbolo de reposo, de duración 1, 1 1/2 ó 2 bits. La recuperación de la información en el equipo receptor se efectúa activando un oscilador local al llegar el bit de comienzo y muestreando con período igual al correspondiente a un bit, guardando en un registro de desplazamiento el valor muestreado. El muestreo del bit de final deberá dar siempre el mismo nivel, por lo que puede utilizarse como detección de carácter con formato inválido.



Fig. 2. Formato de carácter asíncrono.

Los circuitos empleados en los adaptadores usuales incluyen, además, otras funciones programables como: número de bits por cada carácter, número de bits de fin, generación y detección de paridad, etc.

El **método síncrono** utiliza sincronización por bloque en lugar de hacerlo por cada carácter. Los caracteres de información van precedidos de algunos caracteres de sincronización para el bloque, que permitirán poner en fase el oscilador del equipo receptor, tanto de bit como de carácter, para así recuperar la información enviada.

El método síncrono es más eficiente, por utilizar menos símbolos para la sincronización que el método asíncrono, que emplea un bit de comienzo y otro de fin de carácter.

La especificación de todas las características propias de la transmisión se dividen en niveles, para distinguir el diferente rango de funciones que debe satisfacerse.

La ISO ha definido un modelo de referencia para la interconexión de sistemas abiertos (OSI = Open Systems Interconnection). Este modelo consta de 7 niveles, que definen las diferentes actividades a realizar en el proceso del intercambio de información entre dos sistemas en general. Para poder entenderse los sistemas han de estar de acuerdo en cada uno de estos niveles. El propósito de se-



Fig. 3. Diagrama de niveles OSI.

pararlo en niveles es el de poder adaptarse a cualquier situación en el proceso de la conexión, responsabilizando en cada acción al nivel que mejor pueda adaptarse. Veamos una descripción somera de cada uno de los niveles.

Corresponden al nivel 1 o Nivel Físico la definición de las características eléctricas, mecánicas y funcionales de las señales.

Corresponden al nivel 2 o Nivel de Enlace la definición de los caracteres a intercambiar para mantener la conexión entre los equipos. El formato de cada carácter estará definido en el nivel anterior.

El nivel 3 o Nivel de Red se encarga de proporcionar el encaminamiento adecuado dentro de las posibilidades de los circuitos del sistema. Realiza el control de la comunicación.

El nivel 4 o Nivel de Transporte garantiza la conexión transparente entre diferentes sistemas físicos. Se encarga de la apropiada compartición de los recursos de comunicación del sistema.

El nivel 5 o Nivel de Sesión proporciona los mecanismos de organización y estructura de las interacciones entre los procesos de aplicación.

El nivel 6 o Nivel de Presentación independiza los intercambios de información de las diferentes formas de representar los datos.

El nivel 7 o Nivel de Aplicación consistirá en el desarrollo de las actividades propias de la aplicación.

Veremos las características de trabajo de algunos casos de aplicación de los niveles 1 y 2, pues son de interés en la conexión de periféricos a los ordenadores personales. Los demás niveles definidos en la propuesta de ISO para los Sistemas Abiertos especifican las características para permitir la interconexión de equipos para trabajar en una determinada

aplicación utilizando medios públicos y privados de manera concurrente.

Para las aplicaciones relacionadas con ordenadores personales que estamos describiendo solamente se necesita tener en cuenta los dos primeros niveles por lo que se describirá uno de los de uso más común para nivel 1 y un ejemplo de aplicación para nivel 2. Los niveles superiores son realmente necesarios para el caso de emulación de terminal o de mantenimiento de un protocolo específico con el periférico.

■ Configuraciones de conexión

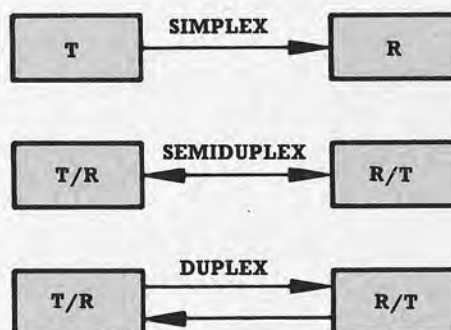


Fig. 4. Modos de conexión.

En la comunicación serie entre equipos pueden considerarse tres modos según las posibilidades de los equipos y la línea de conexión.

— Simplex, es la conexión con comunicación en un solo sentido, como, por ejemplo, la impresora.

— Semiduplex, si la comunicación puede realizarse en ambos sentidos, pero no simultáneamente. La limitación puede ser causada por el medio de comunicación, por ejemplo, la transmisión a dos hilos con modem de una sola portadora.

— Duplex, si la comunicación puede hacerse simultáneamente entre las dos estaciones. Puede realizarse con línea de cuatro hilos o con modem de doble portadora.

Según las necesidades físicas de conexión de uno o más equipos entre sí es posible realizar las conexiones de diferentes maneras:

— Punto-punto. Permite la conexión de dos equipos, generalmente de igual nivel de jerarquía. La comunicación se establece por

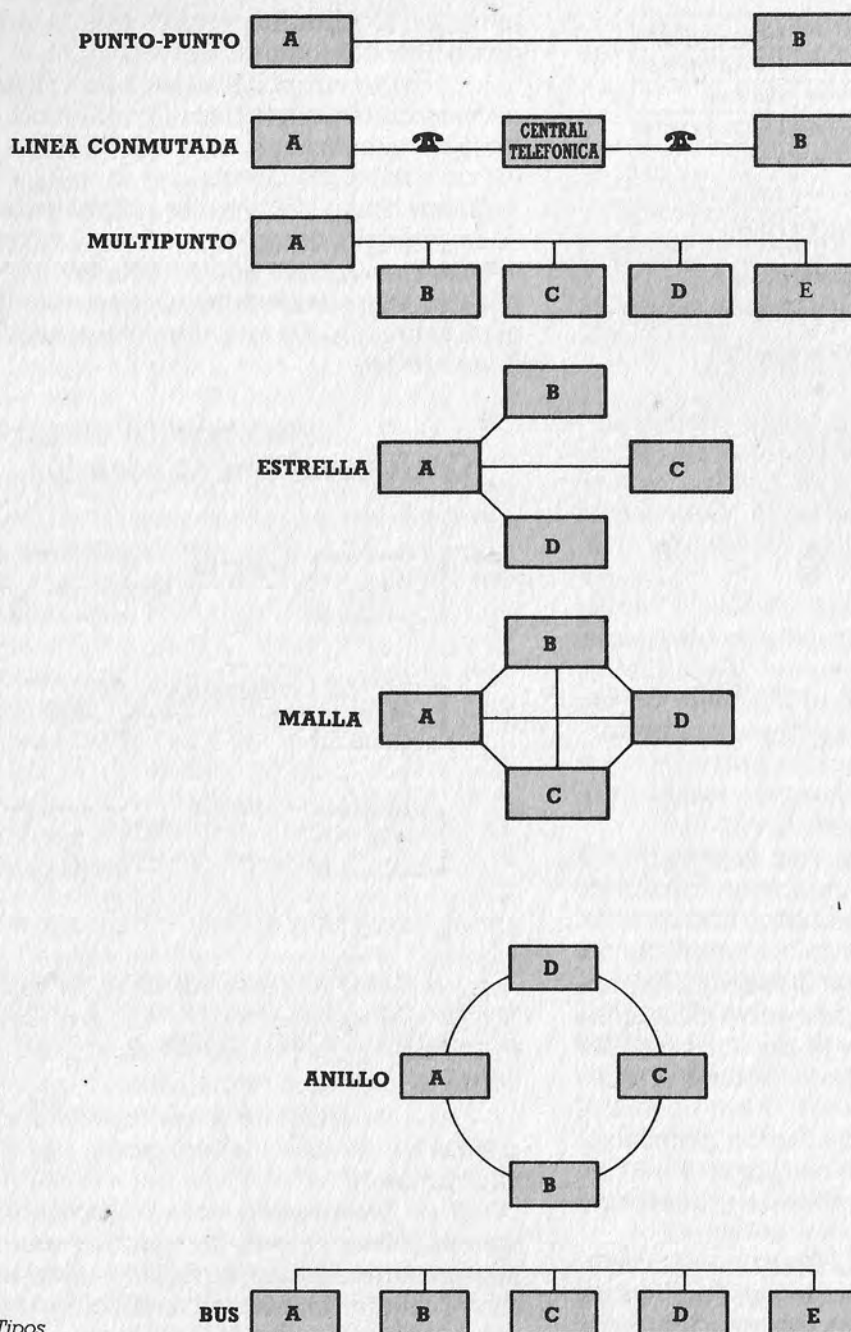


Fig. 5. Tipos de conexión.

envío de petición o directamente según que el medio sea semiduplex o full-duplex.

— Línea conmutada. La conexión se establece a través de la red telefónica mediante llamada manual o automática. Una vez establecido el canal físico la comunicación prosigue como en el caso punto-punto, hasta que se cuelga la línea, con lo que deshace la conexión.

— Multipunto con una estación central. La comunicación es posible entre la unidad

central y cualquiera de las demás estaciones. Periódicamente la estación central sondea a los demás equipos sobre su necesidad de comunicar. Si la respuesta es positiva se establece una comunicación durante el tiempo de un mensaje entre las dos estaciones como si fuera punto-punto.

— Red en estrella. Es un tipo de conexión parecida a la conexión multipunto en la que hay un puerto de control para cada enlace, por lo que pueden realizarse comunica-

ciones simultáneas entre varias estaciones y la estación central.

- Red en malla. Pueden existir conexiones entre todas las estaciones entre sí con lo que se forma una verdadera red. Es la estructura usada por las redes actuales de área amplia, tanto públicas como privadas.

- Red en anillo. La conexión se establece entre cada dos estaciones. La comunicación entre dos estaciones cualesquiera requiere el paso de los mensajes por las estaciones intermedias. Es empleada en redes de área local.

- Red en bus o multipunto. La conexión se hace desde cualquiera de las estaciones a un canal serie común, siendo todas las estaciones de rango equivalente en cuanto a la utilización del medio. Se emplea también en redes de área local.

Pueden darse en algunas circunstancias configuraciones intermedias de conexión, pues el comportamiento lógico puede ser diferente que la distribución física. Así en una red en bus puede efectuarse el control mediante técnica de paso de testigo en secuencia, como si se tratara de un anillo.

■ Protocolos de control de línea

El intercambio de comunicación entre estaciones debe hacerse de forma ordenada de acuerdo con una secuencia predefinida para cada tipo de mensaje. El conjunto de especificaciones que indican la secuencia que debe mantenerse es lo que se denomina protocolo de control de línea. Para cada configuración y cada método de codificación de la información puede establecerse un protocolo diferente o variante de uno dado.

Distinguiremos en los protocolos dos fases de la comunicación: La fase de establecimiento de enlace, en la cual se trata de poner de acuerdo a las estaciones que van a intercambiar información y la fase de transferencia de datos, durante la cual se intercambia información y se lleva el control de errores y sincronización.

Los aspectos más importantes en la fase de conexión son:

- Determinar disponibilidad de la estación receptora.

- Determinar el camino de conexión.

Los aspectos más importantes en la fase de intercambio:

- Configuración del bloque: cabecera, texto, control de error.

- Mantenimiento de la transparencia de la comunicación.

- Control y recuperación de los errores, normalmente se realiza evaluando un código mediante operaciones sobre los datos enviados o recibidos, que se adjunta al mensaje y se comprueba mediante la misma operación que no ha habido alteraciones durante la transmisión. Según el tipo de comunicación podrá pedirse la retransmisión de un bloque o se tratará de recuperar la información original, si el código empleado tiene posibilidad de detección y corrección.

- Verificar la secuencia de transmisión de bloques.

Se describen sucintamente las características fundamentales de los protocolos más usuales. Es de hacer notar que en las comunicaciones serie asíncronas actuales no hay un nivel de estandarización equivalente al alcanzado en los otros métodos, siendo en la mayoría de los casos específicos con las variantes que se indican.

■ Asíncronas

- XON/XOFF. Este protocolo se utiliza en comunicación Duplex, permite el control de las situaciones de saturación de las memorias intermedias del dispositivo receptor. Las estaciones envían mensajes por su canal de transmisión. Cesan la transmisión al recibir el carácter XOFF y la reanudan al recibir XON. La estación receptora envía el carácter XOFF cuando su capacidad de almacenamiento alcanza un nivel determinado, por ejemplo, el 70%. Cuando por efecto de descarga del dispositivo se alcanza un nivel más bajo, por ejemplo, 30%, se envía XON para continuar la comunicación.

- RTS/CTS o DTR. Utilizado en comunicación sin intermedio de modem, a cinco hilos, para controlar el flujo por parte de la estación receptora. Para que se establezca la comunicación el receptor activa su RTS. El transmisor consulta el estado del receptor antes de transmitir.

- ENQ/ACK. Protocolos orientados a utilización de medios semiduplex, en los cuales la estación que desea transmitir solicita la línea mediante el envío del carácter ENQ. Si

EL TALLER DEL HARDWARE

recibe como contestación ACK comienza su transmisión, si recibe NAK, reintenta un número determinado de veces. Utilizan caracteres de control con significado definido en el juego de caracteres ASCII. El protocolo incluye tiempos de espera para el reintento ante la falta de contestación.

— Multipunto con estación de control. Utilizado en conexión en paralelo a una línea semiduplex de varios terminales. Requiere que la unidad central explore la necesidad de comunicación de cada terminal periódicamente. El mensaje de sondeo incluye tras el ENQ el identificador de estación. La estación direccionada contesta afirmativamente si tiene algo que transmitir y negativamente si no lo tiene. La transmisión continúa con el mensaje.

■ Especificaciones de nivel físico

Para los posibles casos de conexión a periféricos existentes interesa concretar las especificaciones para enlace de equipos de comunicación de datos (DCE) y equipo terminal de datos (DTE) o de dos de éstos sin equipo intermedio.

■ RS-232-C, V24

Las especificaciones de RS-232-C del EIA y las de V-24 de CCITT coinciden prácticamente en todos sus valores, por lo que es común considerarlos como idénticos. Es conveniente, sin embargo, aclarar que ambas especifican las características mecánicas de la conexión: tipo de conector, asignación de señales a cada pin del conector. Las características eléctricas son especificadas por la recomendación V-28 del CCITT, estando incluidas en la RS-232-C del EIA.

Los circuitos especificados por estas recomendaciones y que son de interés para la conexión de DTE a un DCE y para nuestro caso son:

PIN CCITT RS-232-C Descripción.

- 1 - Circuito 101 - AA - Tierra de protección.
- 2 - Circuito 103 - BA - TXD - Dato transmitido.
- 3 - Circuito 104 - BB - RXD - Dato recibido.

4 - Circuito 105 - CA - RTS - Petición de envío.

5 - Circuito 106 - CB - CTS - Preparado para transmisión.

6 - Circuito 107 - CC - DSR - Equipo de comunicaciones preparado.

7 - Circuito 102 - AB - GND - Retorno común.

8 - Circuito 109 - CF - DCD - Detección de portadora de línea.

20 - Circuito 108 - CD DTR - Terminal de datos preparado.

22 - Circuito 125 - CE - RI Indicador de llamada.

Las denominaciones de las señales son para el equipo terminal de datos, que se supone está conectado a un equipo de comunicaciones (MODEM). Se han incluido denominaciones de cada circuito según ambas recomendaciones.



Fig. 6. Conexión directa V24.

Para comunicación local con otro ordenador o equipo sin intermedio de Modem puede utilizarse cualquiera de las formas de la figura a 3, 5 ó 7 hilos. El programa en cada uno de los equipos deberá tener en cuenta la conexión exacta para contemplar los casos de error, pues las señales de control y señalización de la interfaz están diseñadas para que se activen como alarmas en cuanto se detecte una alteración. Los errores son, por ejemplo, del tipo de pérdida de señal en la línea, desconexión o avería en el modem, situación de congestión en el otro extremo.

Pero la cosa no es tan simple como parece. En la realidad los diferentes fabricantes han optado por tomar algunos "atajos" para

simplificar la conexión y lo que han conseguido es complicarlo. Veremos en los siguientes tomos algunos casos particulares y cómo solucionarlos, mediante una caja de interconexiones que nos ayude a investigar de qué tipo es nuestra máquina.

Las características mecánicas del conector establecen como norma el conector de 25 pines tipo D de Cannon como el normal, siendo de pines macho para el lado del terminal y pines hembra para el lado del equipo de transmisión. Para conexiones locales, deberá utilizarse un cable especial para realizar los cambios como los indicados en la figura.

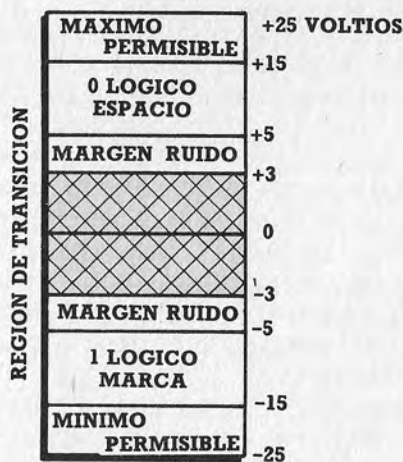


Fig. 7. Niveles de tensión en V24.

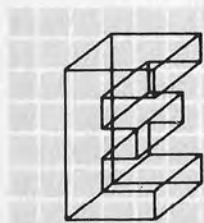
Las características eléctricas de las señales en la interfaz especifican circuitos no balanceados con niveles comprendidos en ± 25 Voltios de acuerdo con el siguiente convenio: < -5 Voltios es nivel 1, $> +5$ Voltios es nivel 0. Los niveles comprendidos entre -3 y $+3$ son considerados como no válidos, por lo que los circuitos de detección de avería se activan al desconectarse cualquiera de los hilos de señal y quedarse al aire. Hay, pues, un margen de ruido de 2 voltios. Los equipos utilizan normalmente tensiones de ± 12 Voltios.

Se especifica también la velocidad máxima de las transiciones, limitándose a 30 Voltios/microsegundo. Esta limitación implica velocidades de transmisión inferiores a 20 Kbps y conexiones a corta distancia (< 12 metros). Para conexiones a muy corta distancia la velocidad puede aumentarse hasta unos 100 Kbps.

También se especifica la impedancia máxima y mínima presentada por los circuitos emisor y receptor, para impedir que cualquier cortocircuito averíe la interfaz. El circuito receptor debe presentar una impedancia de entrada comprendida entre 3 y 7 Kiloohms, mientras el circuito emisor debe presentar una impedancia de salida no superior a 300 ohms, cuando la alimentación esté apagada. Hay circuitos integrados que permiten realizar la interfaz en un solo chip: Para TTL-RS232 el 1488 y para la inversa el 1489.

NATURALEZA Y TECNOLOGIA

■ Cálculo del área de polígonos regulares



N este apartado se ofrece hoy un programa que nos permite resolver un problema que se plantea en numerosos temas relacionados con la Física y la tecnología: la obtención del área de un polígono regular de cualquier número de lados.

Un polígono regular es aquel que cumple la condición de que todos sus lados tienen la misma longitud.

La fórmula para calcular el área de un polígono regular es la siguiente:

$$\text{Area} = \text{Perímetro} \times \text{apotema} / 2$$

en donde el perímetro es la suma de todos sus lados y la apotema la recta que va desde el centro del círculo en el que se encuentra inscrito el polígono al punto medio de cualquiera de sus lados. En la figura siguiente se puede observar para el ejemplo de un pentágono y de un hexágono lo que sería la apotema:

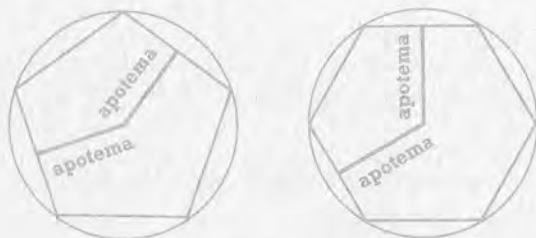


Fig. 1.

De lo expresado en estas figuras anteriores podemos ver que un polígono regular siempre se puede inscribir dentro de una cir-

cunferencia y además también se puede observar que la apotema cumple la condición de que es perpendicular al lado. (Esta condición nos va a resultar muy interesante para los cálculos que vamos a realizar seguidamente.)

El programa que se ofrece en esta sección calcula el área de un polígono regular a partir de los siguientes datos de entrada que se le introducen mediante dos instrucciones INPUT: el número de lados del polígono y la longitud de estos lados (la misma para todos evidentemente).

Para aplicar la fórmula del área hay que conocer, como ya se ha expresado anteriormente, el perímetro y la apotema. El perímetro se obtiene de manera inmediata multiplicando la longitud del lado por el número de lados. La apotema, a partir de los datos que se tienen, se calcula de la siguiente manera:



Fig. 2.

En esta figura podemos ver que la mitad del lado, la apotema y el radio de la circunferencia forman un triángulo rectángulo. Como se puede observar, esta condición se cumple independientemente del número de lados que tenga el polígono. De este triángulo rectángulo son datos conocidos la mitad del lado y el ángulo señalado en el dibujo como

alfa, ya que el ángulo beta que cubre un lado del polígono será igual a los 360° de la circunferencia dividido entre el número de lados del polígono, y, por tanto, alfa será la mitad de beta, por ser el triángulo formado por los dos radios y el lado isósceles.

$$\beta = \frac{360^\circ}{n.^\circ \text{ de lados}}$$

Fig. 3.

Volviendo al triángulo rectángulo, y siendo ya datos conocidos en él la mitad del lado (un cateto) y el ángulo alfa, podemos cal-

cular la apotema (el otro cateto) muy fácilmente a partir de la fórmula de la tangente:

$$\operatorname{tg} \alpha = \frac{\text{Lado}/2}{\text{apotema}} \Rightarrow \text{apotema} = \frac{\text{Lado}/2}{\operatorname{tg} \alpha}$$

Fig. 4.

Una vez calculado el perímetro y la apotema, ya podemos aplicar la fórmula del área y hallar ésta.

Explicación del programa

El programa que realiza el problema planteado es el siguiente:

```

10 REM *****
20 REM **
30 REM **   PROGRAMA PARA CALCULAR EL AREA DE POLIGONOS REGULARES   **
40 REM **   **
50 REM *****
60 CLS
70 INPUT "INTRODUCE EL NUMERO DE LADOS";L
80 INPUT "¿CUAL ES LA LONGITUD DEL LADO?";LADO
90 LET PI=3.141596
100 LET A=2*PI/(L*2)
110 LET AP=(LADO/2)/TAN(A)
120 LET AR=(LADO*L)*AP/2
130 PRINT:PRINT:PRINT "EL AREA ES";AR
140 PRINT:PRINT "¿QUIERES QUE TE DIBUJE EL POLIGONO? (S/N)"
150 LET A$=INKEY$
160 IF A$="" THEN GOTO 150
170 IF A$="N" THEN END
180 IF A$<>"S" THEN GOTO 150
190 LET R=(LADO/2)/SIN(A)
195 REM La instruccion 200 solo se debe poner para IBM o compatibles y MSX
200 SCREEN 1
210 CLS
220 INPUT "INTRODUCE LAS COORDENADAS DEL CENTRO DEL POLIGONO";C1,C2
230 CLS
240 LET P=2*PI/L
250 LET X1=C1+R
260 LET Y1=C2
270 PSET(X1,Y1),2
280 FOR I=P TO (2*PI)+.02 STEP P
290 LINE-(C1+(R*COS(I)),C2+(R*SIN(I))),2
300 NEXT I
310 REM *****
320 REM **   MODIFICACIONES PARA EL SPECTRUM:   **
330 REM **   170 IF A$="N" THEN GOTO 9999   **
340 REM **   270 PLOT X1,Y1   **
350 REM **   290 DRAW C1+(R*COS(I))-X1,C2+(R*SIN(I))-Y1   **
360 REM **   295 LET X1=C1+(R*COS(I)):LET Y1=C2+(R*SIN(I))   **
370 REM **   MODIFICACIONES PARA COMMODORE:   **
380 REM **   Sustituir en 60,210,230 el CLS por PRINT CHR$(147)   **
390 REM **   150 GET A$   **
400 REM **   270 PLOT X1,Y1,1   **
410 REM **   290 LINE X1,Y1,C1+(R*COS(I)),C2+(R*SIN(I)),1   **
420 REM **   295 LET X1=C1+(R*COS(I)):LET Y1=C2+(R*SIN(I))   **
430 REM **   MODIFICACIONES PARA EL AMSTRAD   **
440 REM **   270 PLOT X1,Y1   **
450 REM **   290 DRAW C1+(R*COS(I)),C2+(R*SIN(I))   **
460 REM *****

```

La primera parte del programa realiza el cálculo del área siguiendo los pasos vistos anteriormente: pide el número de lados y la longitud de éstos, halla el valor del ángulo alfa (en el programa representado por la variable A), halla el valor de la apotema y, finalmente, el del área escribiéndolo por pantalla. De esta parte, conviene únicamente llamar la atención de que debido a que el ordenador opera en radianes, el valor del ángulo se calcula en éstos.

El resultado del área obtenido saldrá siempre con un pequeño error por defecto del orden de 0.0001 debido a que tanto el valor de pi como el de la tangente del ángulo que calcula el programa tienen que quedar evidentemente truncados al número de decimales con el que opere nuestro ordenador.

El programa tiene una segunda parte (a partir de la instrucción 150) en la que ofrece al usuario la posibilidad de realizar el dibujo

APRENDER CON EL ORDENADOR

del polígono del que se está calculando el área. Si se teclea la letra S (opción de dibujar la figura), el programa pedirá las coordenadas del punto que se desee que sea el centro del polígono y dibujará éste, teniendo en cuenta que tomará para dibujar cada lado tantos puntos, de los correspondientes a la división de la pantalla, como longitud se haya introducido en el apartado anterior, por lo que **únicamente se podrá realizar** la opción de dibujar el polígono cuando la dimensión de los lados esté dentro de los límites de las coordenadas de la pantalla de gráficos de nuestro ordenador.

En la figura siguiente observamos cómo quedaría un hexágono dibujado en pantalla con este programa:

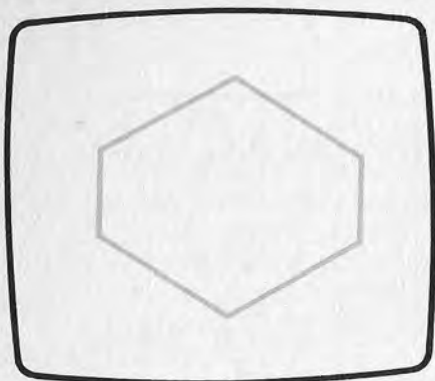


Fig. 5.

MATEMATICAS

■ Cálculo del determinante de una matriz de cualquier rango

Este programa que se ofrece en este apartado realiza el cálculo del determinante de una matriz de cualquier rango.

Recordamos que el método general para calcular el determinante de una matriz es el sumatorio de los productos de los elementos de una fila o columna cualquiera por sus adjuntos correspondientes.

El adjunto del elemento genérico (i, j) será el determinante de la matriz resultante de eliminar a la matriz inicial la fila i y la columna j . De todas maneras, hay que destacar que el "sumatorio" indicado anteriormente no es lo que matemáticamente se entiende por sumatorio, ya que los productos de los elementos (i, j) que cumplan la condición de que $i + j$ sea número impar deben ser restados mientras que el resto se deben sumar.

A continuación podemos ver esta definición en forma de fórmula matemática y cómo se aplicaría a un ejemplo concreto de una matriz de 4×4 .

$$D = a_{11} |Adj|_{11} - a_{12} |Adj|_{12} + a_{13} |Adj|_{13} - \dots + a_{1n} |Adj|_{1n}$$

Siendo i n.º impar

$$A = \begin{vmatrix} 7 & 4 & 2 & 9 \\ 3 & 5 & 4 & 1 \\ 2 & 3 & 8 & 3 \\ 9 & 9 & 2 & 1 \end{vmatrix} \rightarrow D = 7 \begin{vmatrix} 5 & 4 & 1 \\ 3 & 8 & 3 \\ 9 & 2 & 1 \end{vmatrix} - 4 \begin{vmatrix} 3 & 4 & 1 \\ 2 & 8 & 3 \\ 9 & 2 & 1 \end{vmatrix} +$$

$$+ 2 \begin{vmatrix} 3 & 5 & 1 \\ 2 & 3 & 3 \\ 9 & 9 & 1 \end{vmatrix} - 9 \begin{vmatrix} 3 & 5 & 4 \\ 2 & 3 & 8 \\ 9 & 9 & 2 \end{vmatrix}$$

Fig. 6.

Los llamados adjuntos son, a su vez, como puede advertirse, determinantes de matrices de rango inferior, y su cálculo se haría aplicando la misma fórmula hasta que se llegara a determinantes de matrices de 2×2 cuya fórmula es:

$$D = a_{11} \times a_{22} - a_{21} \times a_{12}$$

Fig. 7.

■ Explicación del programa

Este es el programa que resuelve este problema:

```

10 REM *****
20 REM **
30 REM **   PROGRAMA PARA CALCULAR EL DETERMINANTE DE UNA MATRIZ   **
40 REM **               DE CUALQUIER RANGO                           **
50 REM **
60 REM *****
70 CLS
80 INPUT "INTRODUCE LA DIMENSION DE LA MATRIZ";N
90 IF N<1 THEN GOTO 80
100 DIM D(N,N):DIM P(N,N,N):DIM S(N*2)
110 FOR I=1 TO N
120 PRINT "INTRODUCE A CONTINUACION LOS NUMEROS DE LA FILA";I

```

```

130 FOR J=1 TO N
140 INPUT D(I,J)
150 NEXT J:NEXT I
160 IF N=1 THEN LET DET=D(1,1):GOTO 260
170 CLS
180 FOR I=1 TO N
190 FOR J=1 TO N
200 PRINT TAB(J*5);D(I,J);
210 NEXT J
220 PRINT :NEXT I
230 LET RANG=N
240 IF N>4 THEN PRINT :PRINT "¡UN MOMENTO!,ESTOY HACIENDO LOS CALCULOS"
250 GOSUB 1000
260 PRINT :PRINT :PRINT "EL DETERMINANTE DE LA MATRIZ ES ";DET
270 END
1000 LET DET=0
1010 IF RANG=2 THEN GOTO 1250
1020 FOR I=1 TO RANG
1030 FOR J=1 TO RANG
1040 LET P(RANG,I,J)=D(I,J)
1050 NEXT J:NEXT I
1060 FOR I=1 TO RANG
1070 FOR J=2 TO RANG
1080 LET COL=1
1090 FOR K=1 TO RANG
1100 IF K>I THEN LET D(J-1,COL)=P(RANG,J,K):LET COL=COL+1
1110 NEXT K:NEXT J
1120 LET RANG=RANG-1
1130 LET SP=SP+1
1140 LET S(SP)=DET
1150 LET SP=SP+1:LET S(SP)=I
1160 GOSUB 1000
1170 LET I=S(SP):LET SP=SP-1
1180 LET RANG=RANG+1
1190 IF (I MOD 2)=1 THEN LET S(SP)=S(SP)+(P(RANG,1,I)*DET)
1200 IF (I MOD 2)=0 THEN LET S(SP)=S(SP)-(P(RANG,1,I)*DET)
1210 LET DET=S(SP)
1220 LET SP=SP-1
1230 NEXT I
1240 RETURN
1250 DET=(D(1,1)*D(2,2))-(D(1,2)*D(2,1))
1260 RETURN
2000 REM *****
2010 REM ** MODIFICACION PARA COMMODORE **
2020 REM ** Sustituir en 70 y 170 CLS por PRINT CHR$(147) **
2030 REM ** MODIFICACION PARA SPECTRUM **
2040 REM ** Sustituir en 270 END por GOTO 9999 **
2050 REM *****

```

El programa comienza pidiendo los datos de la matriz de la que se quiere hallar el determinante y guardando ésta en la tabla D. A continuación escribe en pantalla la matriz introducida y comienza el cálculo de su determinante, de lo que se encarga la subrutina 1000, escribiendo después su valor que se encuentra en la variable DET, que es donde la subrutina hace el cálculo de éste.

La subrutina realiza el cálculo del determinante de la matriz que se encuentra almacenada en la tabla D y cuyo rango está indicado en la variable RANG. Para este cálculo, la subrutina emplea lo que se llaman las técnicas recursivas, que quieren decir que la misma subrutina, cuando para calcular el determinante que está realizando, necesita calcular otro determinante (tal y como exige la fórmula), se hace una llamada a sí misma para que calcule este otro determinante.

Veamos, a grandes rasgos, una idea general de estas técnicas en la subrutina de nuestro programa. La subrutina comienza inicializando a cero la variable DET, en la que va a calcular el resultado del determinante, y a continuación comprueba si el rango de éste es 2. En caso de que sea 2, salta a la línea 1250

en la que resuelve el determinante por la fórmula correspondiente a los determinantes de 2×2 .

Si el rango del determinante que queremos calcular fuera mayor de 2, entonces habría que aplicar la fórmula general vista anteriormente. La subrutina aplica siempre esta fórmula con los elementos de la primera fila. Como para aplicar esta fórmula habrá que calcular una serie de determinantes, la subrutina se hará una llamada a sí misma para calcular cada uno de estos determinantes.

Cada vez que la subrutina se haga una llamada a sí misma deberá poner en la tabla D el determinante que se quiere calcular y, por tanto, antes de hacer esto debe guardar en otra tabla lo que tiene en la tabla D para que no se pierda al guardar el nuevo determinante. Esta tabla donde se van guardando los determinantes de los distintos rangos que se van calculando es la tabla de tres dimensiones P y entre las instrucciones 1020 y 1050 se guarda en ella el determinante que está calculando la subrutina antes de llamarse a sí misma para hallar los determinantes que necesite.

APRENDER CON EL ORDENADOR

$$A = \begin{vmatrix} 4 & 2 & 9 & 1 \\ 3 & 4 & 3 & 2 \\ 2 & 9 & 8 & 7 \\ 3 & 4 & 3 & 3 \end{vmatrix}$$

$$P(4, N, N) = \begin{vmatrix} 4 & 2 & 9 & 1 \\ 3 & 4 & 3 & 2 \\ 2 & 9 & 8 & 7 \\ 3 & 4 & 3 & 3 \end{vmatrix}$$

$$P(3, NN) = \begin{vmatrix} 4 & 3 & 2 \\ 8 & 8 & 7 \\ 4 & 3 & 3 \end{vmatrix}$$

$$P(3, N, N) = \begin{vmatrix} 3 & 3 & 2 \\ 2 & 8 & 2 \\ 3 & 3 & 3 \end{vmatrix}$$

$$P(3, N, N) = \begin{vmatrix} 3 & 4 & 2 \\ 2 & 9 & 7 \\ 3 & 4 & 3 \end{vmatrix} - \begin{vmatrix} 3 & 4 & 3 \\ 2 & 9 & 8 \\ 3 & 4 & 3 \end{vmatrix}$$

Fig. 8. Valores que irán teniendo para este ejemplo, las distintas tablas de P, mientras la subrutina calcula los determinantes de rango 2 que habría que hallar para resolver cada uno de los 4 determinantes de rango 3 a los que se ha llegado.

La instrucción 1060 es un FOR dentro del cual se va a calcular cada uno de los sumandos del "sumatorio" de la fórmula. Las instrucciones 1070 a 1110 obtienen el adjunto (determinante) que hay que calcular guardándolo en la tabla D. A continuación se llama a la subrutina en 1160 para calcular este determinante guardando previamente en la lista auxiliar S los valores de I y DET para que no los perdamos al volver a comenzar de nuevo la subrutina.

Las instrucciones 1170 a la 1220 son las encargadas de sumar o restar el producto del elemento de la fila correspondiente por el adjunto calculado en la subrutina de 1160, recuperando también los valores guardados en la lista S anteriormente.

En estas líneas anteriores se ha tratado de que te hagas una idea muy general del funcionamiento de la técnica recursiva empleada, aunque si nunca has programado con técnicas recursivas es posible que no hayas llegado a comprender todos los detalles más concretos de estas técnicas, ya que éstas son precisamente de los métodos de programación más difíciles y una explicación detallada de ellos llevaría varios capítulos de un libro.

NOTA: Conviene destacar que debido al gran número de operaciones que hay que realizar para el cálculo de los determinantes de rango mayor que 4, como tú ya sabrás si alguna vez has tenido que resolver alguno de ellos manualmente, cuando ejecutamos el pro-

grama para calcular alguno de éstos el resultado no nos lo proporciona automáticamente, sino que tarda un rato, que según el rango puede ser de medio minuto, un minuto o más. Por tanto, si cuando ejecutas el programa para algún determinante de un rango grande ves que no te escribe el resultado, espera y no rompas el programa, porque, aunque tarde un rato, te obtendrá el resultado.

SOCIEDAD

Test sobre ríos del mundo

Con el programa que te ofrecemos a continuación, vas a tener la oportunidad de aprender nombres de ríos del mundo a través de un entretenido juego.

El programa te escribirá en pantalla una frase con las características del río sobre el que te está preguntando y, a continuación, escribirá la primera y la última letra de su nombre, así como el número de letras que faltan. Seguidamente, tú deberás ir tecleando las letras que creas que faltan para completar el nombre del río. Cuando teclees una letra que está en el nombre, el programa te la colocará en su lugar adecuado y deberás continuar hasta que completes el nombre. En la siguiente figura podemos observar una imagen de cómo tendríamos la pantalla en un momento determinado de la ejecución mientras estamos intentando adivinar el nombre.



Fig. 9.

El sistema es el mismo que el del popular juego del "ahorcado".

Explicación del programa

El esquema general del programa es el mismo que otro publicado en un número ante-

rior que realizaba este mismo juego, aunque en este caso evidentemente varían los datos

correspondientes a las palabras y definiciones que hay que adivinar (ríos del mundo).

```

10 REM *****
20 REM **
30 REM ** TEST SOBRE RIOS DEL MUNDO **
40 REM **
50 REM *****
60 DIM P$(20,11):DIM L$(11)
70 REM LA INSTRUCCION 100 SOLO ES NECESARIO UTILIZARLA PARA MS-DOS (IBM O
   COMPATIBLES)
80 RANDOMIZE TIMER
90 DATA 80,69,67,72,79,82,65,0,68,65,78,85,66,73,79,0,79,66,73,0,84,65,77,69
100 DATA 83,73,83,0,68,78,73,69,80,69,82,0,82,79,68,65,78,79,0,76,79,73,82,65
110 DATA 0,69,76,66,65,0,77,65,67,75,69,78,90,73,69,0,83,65,78,76,79,82,69,78
120 DATA 90,79,0,65,82,75,65,78,83,65,83,0,79,82,73,78,79,67,79,0,65,77,65,90
130 DATA 79,78,65,83,0,77,73,83,83,79,85,82,73,0,65,90,85,76,0,78,73,76,79,0
140 DATA 86,73,83,84,85,76,65,0,84,73,66,69,82,0,80,65,82,65,78,65,0,77,65
150 DATA 68,69,73,82,65,0
160 PRINT :PRINT "¡UN MOMENTO!, ESTOY CARGANDO LOS DATOS"
170 FOR I=1 TO 20
180 LET J=1
190 READ A
200 IF A=0 THEN FOR K=J TO 11 :LET P$(I,K)=" ":NEXT K :GOTO 230
210 LET P$(I,J)=CHR$(A)
220 LET J=J+1:GOTO 190
230 NEXT I
240 LET PAL=INT(RND(1)*20)+1
250 FOR J=1 TO 11 :LET L$(J)=" ":NEXT J
260 LET FT=0 :LET CON=0
270 CLS
280 ON PAL GOTO 290,300,310,320,330,340,350,360,370,380,390,400,410,420,430,440,
   450,460,470,480
290 PRINT "RIO QUE NACE EN LOS MONTES URALES Y DESEMBOCA EN EL OCEANO GLACIAL AR
   TICO":GOTO 490
300 PRINT "RIO QUE ATRAVIESA TODO EL CENTRO DE EUROPA Y QUE ES FAMOSO PORQUE SE
   DICE QUE LOS ENAMORADOS LO VEN DE COLOR AZUL":GOTO 490
310 PRINT "RIO QUE ATRAVIESA SIBERIA":GOTO 490
320 PRINT "RIO QUE PASA POR LONDRES":GOTO 490
330 PRINT "RIO RUSO QUE PASA POR LA CIUDAD DE KIEV":GOTO 490
340 PRINT "PRINCIPAL RIO FRANCES DE LOS QUE DESEMBOCAN EN EL MEDITERRANEO" :
   GOTO 490
350 PRINT "RIO FRANCES CONOCIDO POR LOS CASTILLOS ENCLAVADOS A LO LARGO DE SU CA
   UCE":GOTO 490
360 PRINT "RIO QUE PASA POR HAMBURGO":GOTO 490
370 PRINT "RIO AMERICANO QUE NACE EN LAS MONTAÑAS ROCOSAS Y DESEMBOCA EN EL OCEA
   NO GLACIAL ARTICO":GOTO 490
380 PRINT "RIO AMERICANO QUE NACE EN EL LAGO ONTARIO Y DESEMBOCA FRENTE A LA ISL
   A DE TERRANOVA":GOTO 490
390 PRINT "AFLUENTE DEL MISSISSIPPI":GOTO 490
400 PRINT "PRINCIPAL RIO DE VENEZUELA":GOTO 490
410 PRINT "¿CUAL ES EL RIO MAS CAUDALOSO DEL MUNDO?":GOTO 490
420 PRINT "RIO DE ESTADOS UNIDOS CUYA LONGITUD ALCANZA LOS 6000 KM":GOTO 490
430 PRINT "RIO CHINO QUE DESEMBOCA CERCA DE SHANGHAI":GOTO 490
440 PRINT "RIO AFRICANO QUE DESEMBOCA EN EL MEDITERRANEO EN CUYO CAUCE SE HIZO L
   A PRESA DE ASUAN":GOTO 490
450 PRINT "RIO QUE PASA POR VARSOVIA":GOTO 490
460 PRINT "RIO QUE PASA POR ROMA":GOTO 490
470 PRINT "RIO QUE SIRVE DE FRONTERA ENTRE PARAGUAY Y ARGENTINA":GOTO 490
480 PRINT "AFLUENTE DEL AMAZONAS":GOTO 490
490 LET L$(1)=P$(PAL,1)
500 FOR I=11 TO 2 STEP-1
510 IF P$(PAL,I)<>" " THEN LET L$(I)=P$(PAL,I):LET F=I:GOTO 530
520 NEXT I
530 LOCATE 10,10
540 FOR I=1 TO F
550 PRINT L$(I); " ";
560 NEXT I
570 PRINT:PRINT TAB(12);
580 FOR I=2 TO F-1
590 PRINT " ";
600 NEXT I
610 REM PARA AMSTRAD :560 LOCATE 1,15
620 LOCATE 15,1
630 IF FT THEN GOTO 750
640 LET FT=1
650 PRINT "TECLEA LAS LETRAS QUE CREAS QUE FALTEN EN LA PALABRA(MAYUSCULAS)"
660 LET A$=INKEY$
670 IF A$="" THEN GOTO 660
680 FOR I=2 TO F-1
690 IF P$(PAL,I)=A$ THEN LET L$(I)=P$(PAL,I):LET LT=1
700 IF L$(I)=A$ THEN LET FT=0
710 NEXT I
720 IF LT=0 THEN CON=CON+1:PRINT:PRINT"LO SIENTO,LA LETRA ";A$;" NO ESTA"
730 LET LT=0
740 GOTO 530
750 PRINT "
760 PRINT :PRINT "
770 PRINT "¡ENHORABUENA!, LA HAS ACERTADO":PRINT
780 PRINT "HAS TECLEADO ";CON;" LETRAS ERRONEAS AL INTENTAR ACERTAR LA PALABRA"
790 PRINT "QUIERES JUGAR OTRA VEZ? (S/N)"
800 LET A$=INKEY$
810 IF A$="S" THEN GOTO 240
820 IF A$<>"N" THEN GOTO 800
830 REM *****
840 REM ** MODIFICACIONES PARA COMMODORE : **
850 REM ** 270 PRINT CHR$(147) **
860 REM ** 530 FOR I=1 TO 8 **
870 REM ** 532 PRINT **
880 REM ** 534 NEXT I :PRINT TAB(10); **
890 REM ** 620 FOR I=1 TO 4 **
900 REM ** 623 PRINT **
910 REM ** 625 NEXT I **
920 REM ** 660 GET A$ **

```

APRENDER CON EL ORDENADOR

```
930 REM ** 740 GOTO 270 **
940 REM ** 800 GET A$ **
950 REM ** MODIFICACIONES PARA SPECTRUM **
960 REM ** 280 GOTO (PAL*10)+280 **
970 REM ** 530 PRINT AT 10,10 **
980 REM ** 620 PRINT AT 15,1 **
990 REM *****
```

El programa comienza con 7 instrucciones DATA en las que se introducen los nombres de los ríos. Como se observará a la vista del programa, los datos de estas instrucciones no se parecen lo más mínimo a una lista de nombres, sino que son una lista ordenada de números. Estos números precisamente se corresponden con los códigos internos (códigos ASCII) que las letras tienen dentro de nuestro ordenador. Cada número es una letra. La razón de introducir en los DATA los códigos de las letras en vez de las propias letras directamente es obvia: si introdujésemos ahí los nombres de los ríos, al teclear el programa en nuestro ordenador tendríamos que teclear estos nombres y cuando posteriormente lo ejecutásemos ya sabríamos estos nombres, con lo que el juego perdería su interés. En las instrucciones DATA, al final de los números correspondientes a las letras de cada una de las palabras, se ha colocado un cero para que el programa pueda localizar cuando los lea, el final de cada palabra.

El programa guarda las palabras que lee, después de transformar los códigos en letras mediante la función CHR\$ en la tabla P\$ (instrucciones 170-230). Hay 20 nombres de ríos en total y el máximo de letras que pueden tener es 11. En los ríos que tienen menos de 11 letras se rellenan con blancos los espacios sobrantes.

En la instrucción 240 el programa selecciona un número al azar entre 1 y 20, que será el número del río que habrá que adivinar. Entonces escribe el texto correspondiente mediante la instrucción ON (línea 280).

El programa guarda en la lista L\$ la palabra con las letras que tú ya has adivinado teniendo blancos en los espacios correspondientes a las letras que aún te quedan por acertar. Así, inicialmente entre las sentencias 490 y 520 guarda en L\$ la primera y última letra de la palabra en sus lugares correspondientes.

Entre las instrucciones 530 y 600 se escriben en la pantalla la letra inicial y final, un guión por cada letra que hay que adivinar, así como todas aquellas letras que haya en L\$ que serán las que hayamos acertado ya, cuando se ejecuten estas instrucciones de nuevo.

Cuando introduces una letra (que se guarda en A\$ en la instrucción 660) se comprueba si está, y si es así se guarda en su lugar correspondiente de L\$ (instrucciones 660-710) para luego volver a 530 donde escribe de nuevo L\$ con las letras introducidas.

Significado de las variables de control que utiliza el programa:

LT = → Indica que la última letra introducida no está en la palabra.

LT = → La última letra introducida sí está en la palabra.

FT = → Indica que ya se ha acertado la palabra.

FT = → La palabra aún no se ha acertado.

F → Número de letras de la palabra (río) que se está adivinando en cada momento.

Las variables LT y FT se ponen a 0 y 1, respectivamente, cada vez que empezamos una nueva operación (líneas 640 y 730) y si luego se comprueba que lo que indican estos valores no se cumple, se les cambia de valor.

NOTA: El programa, como ya se ha dicho anteriormente, selecciona de manera aleatoria el río que hay que adivinar, por lo que puede darse el caso de que a veces se repita un mismo río dos veces casi seguidas, pero, sin embargo, esta técnica de no seguir un orden preestablecido aumenta el interés del juego, ya que cuando ejecutes el programa de nuevo en otras ocasiones no sabrás qué río o ríos te van a salir.

PARA LOS MAS JOVENES

■ Divertidas preguntas de ingenio

Os presentamos en este apartado un programa que plantea curiosas y divertidas preguntas de ingenio que os pueden servir para pasar un rato entretenido con vuestros compañeros y para ver cómo muchas cosas que habéis estudiado en los libros de texto tienen divertidas aplicaciones en casos de la vida real.

El programa, después de darle la orden de ejecución, escribe en pantalla la primera pregunta y a continuación te da tres opciones (para cada una de las cuales tienes que pulsar la tecla que se te indica): conocer la respuesta a la pregunta, que te escriba la siguiente pregunta o que termine el programa. En la siguiente figura podemos observar un ejemplo de lo que nos saldría en la pantalla cada vez que nos plantea una pregunta:

En total, hay 18 preguntas y si, de las opciones anteriores, eliges la correspondiente a conocer la respuesta, después de escribirte ésta te vuelve a dar la opción de continuar con la siguiente pregunta o terminar.



Fig. 10.

```

10 REM *****
20 REM **
30 REM **      PROGRAMA QUE PLANTEA DIVERTIDAS PREGUNTAS DE INGENIO      **
40 REM **
50 REM *****
60 DIM N$(85)
70 DATA "DE LA TIERRA " "ANA", "HICIERAS TE APLAS" "50). LA DISTANCIA " "36 ("
80 DATA "EN FLU" "BRO" "FAL" "PORQUE ESTA EN " "PORQUE " "DEL VATI" "6x6)"
90 DATA "TON YA QUE AL SER EL MAS " "KM CUADRADOS" "SA, PORQUE ENTONCES "
100 DATA "PAJA QUE EN EL " "5" "CIUDAD " "AL SOL SU " "SABER QUE " "2"
110 DATA "10 ES " "DESPUES" "MECHA" "+4+4=100" "LLAR" "UN BI" "CANO-->0.5 "
120 DATA "NO SE PODIA " "LEJANO " "MO, YA QUE SEGUN EL " "EN BASE" "AL SOL " "4"
130 DATA "SIEMPRE VEMOS " "E" "PRINCIPIO DE " "MILLONES" (1.000) "AGUA EN EL " "4"
140 DATA "ALLI ES MAYOR " "M {2 ELEVADO A " "COMO EL ACEITE " "248 DE LOS "
150 DATA "QUINQUE " "LA MISMA PORQUE " "EL POLO NORTE " "LLON, QUE ES UN "
160 DATA "JO" "HASTA ALCANZAR LA " "GIRA ALREDEDOR " "AGUA, EL ACEITE FLO"
170 DATA "NUESTROS" "LA LUNA " "PERO NO " "CRISTO " "1.000.000.000" "PESARIA 1"
180 DATA "MEDES, EL EMPUJE HACIA ARRIBA " "GIRA SOBRE " "AÑO DURA " "ECHANDO "
190 DATA "GUADI", "TARIA Y SUBIRIA " "LA GRAVEDAD" "K" "99 AÑOS " "TA"
200 DATA "MILLON DE " "PESA " "MAS VOLUMEN " "50 KILOS, YA QUE " "EL DE PLO"
210 DATA "PLOMO, AL OCUPAR " "SI MISMA", "TARIA PORQUE EN JUPITER " "4+4"
220 DATA "ES MAYOR EN LA " "ARQUI", "SI LO " "ARRU", "NACERIA " "MENOS QUE EL "
230 DATA ".900.000 " "1.125", " 2"
240 CLS
250 FOR I=1 TO 85
260 READ N$(I)
270 NEXT I
280 PRINT "UNA PERSONA ESTA EN UN LUGAR MIRANDO HACIA EL SUR Y SE DA MEDIA VUELTA
A Y SIGUE MIRANDO HACIA EL SUR ¿COMO ES ESTO POSIBLE?"
290 GOSUB 2000
300 IF RES=0 THEN GOTO 330
310 PRINT :PRINT N$(9);N$(46)
320 GOSUB 2030
330 CLS
340 PRINT "¿QUE PESA MAS, UN KILO DE PLOMO O UN KILO DE PAJA?"
350 GOSUB 2000
360 IF RES=0 THEN GOTO 390
370 PRINT N$(72);N$(31);N$(37);N$(78);N$(58);N$(77);N$(16);N$(73);N$(70)
380 GOSUB 2030
390 CLS
400 PRINT "EN UNA EXCAVACION ARQUEOLOGICA SE ENCUENTRA UNA MONEDA CON LA INSCRIP
CION: 'ROMA, AÑO 99 ANTES DE CRISTO'. ¿ES VERDADERA O FALSA?"
410 GOSUB 2000
420 IF RES=0 THEN GOTO 450
430 PRINT N$(8);N$(15);N$(29);N$(20);N$(55);N$(81);N$(66);N$(23)
440 GOSUB 2030
450 CLS
460 PRINT "¿CUAL ES EL RIO MAS LARGO DE ESPAÑA?"
470 GOSUB 2000
480 IF RES=0 THEN GOTO 510
490 PRINT N$(36);N$(7)
500 GOSUB 2030
510 CLS
520 PRINT "¿Y EL RIO MAS LARGO DE LA PENINSULA IBERICA?"
530 GOSUB 2000
540 IF RES=0 THEN GOTO 570
550 PRINT N$(67);N$(48)
560 GOSUB 2030
570 CLS
580 PRINT "¿CUAL ES EL PAIS MAS PEQUEÑO DEL MUNDO Y QUE EXTENSION TIENE?"
590 GOSUB 2000
600 IF RES=0 THEN GOTO 630
610 PRINT N$(18);N$(11);N$(28);N$(14)
620 GOSUB 2030
630 CLS
640 PRINT "2+2=10. ¿COMO ES ESTO POSIBLE?"
650 GOSUB 2000
660 IF RES=0 THEN GOTO 690
670 PRINT N$(10);N$(22);N$(34);N$(32);N$(85)
680 GOSUB 2030
690 CLS
700 PRINT " SI TENEMOS UN PAPEL DE 1 MILMETRO DE GROSOR Y LE DOBLAMOS POR LA MI
TAD,
TAD,
VEZ,
OBTENDRAMOS UN GROSOR DE 2 MILMETROS. REPITIENDO LA OPERACION OTRA
EL PAPEL TENDRIA UN GROSOR DE 4 MILIMETROS"

```

APRENDER CON EL ORDENADOR

```

710 PRINT "¿QUE GROSOR TENDRIA EL PAPEL SI LE DOBLAMOS 50 VECES?"
720 GOSUB 2000
730 IF RES=0 THEN GOTO 760
740 PRINT N$(84);N$(83);N$(65);N$(41);N$(4);N$(1);N$(33)
750 GOSUB 2030
760 CLS
770 PRINT "¿NOS MUESTRA LA LUNA SIEMPRE SU MISMA CARA?"
780 GOSUB 2000
790 IF RES=0 THEN GOTO 820
800 PRINT N$(35);N$(45);N$(53);N$(50);N$(1);N$(54);N$(59);N$(74)
810 GOSUB 2030
820 CLS
830 PRINT "¿QUE ES MAYOR: UN BILLON O MIL MILLONES?"
840 GOSUB 2000
850 IF RES=0 THEN GOTO 880
860 PRINT N$(27);N$(47);N$(68);N$(38);N$(56)
870 GOSUB 2030
880 CLS
890 PRINT "¿PERMITIRIAS QUE SE TE SENTASE SOBRE LAS RODILLAS UNA PERSONA DE 60 K
      IL05 EN EL PLANETA JUPITER?"
900 GOSUB 2000
910 IF RES=0 THEN GOTO 940
920 PRINT N$(79);N$(3);N$(75);N$(57);N$(71);N$(40);N$(64)
930 GOSUB 2030
940 CLS
950 PRINT "¿CUAL ES EL SIGUIENTE NUMERO DE ESTA SERIE:?"
960 PRINT "10-20-15-25-20-30"
970 GOSUB 2000
980 IF RES=0 THEN GOTO 1010
990 PRINT N$(21);N$(17)
1000 GOSUB 2030
1010 CLS
1020 PRINT "¿EN QUE PLANETA SERIA MAS LARGA UNA CONDENA DE UN AÑO DE CARCEL?"
1030 GOSUB 2000
1040 IF RES=0 THEN GOTO 1070
1050 PRINT N$(6);N$(13);N$(30);N$(19);N$(60);N$(43);N$(52)
1060 GOSUB 2030
1070 CLS
1080 PRINT "¿COMO COLOCARIAS SIETE CUATROS DE FORMA QUE SUMASEN 100?"
1090 GOSUB 2000
1100 IF RES=0 THEN GOTO 1130
1110 PRINT N$(34);N$(76);N$(76);N$(25)
1120 GOSUB 2030
1130 CLS
1140 PRINT "PERRO ES A LADRAR COMO PALOMA ES A..."
1150 GOSUB 2000
1160 IF RES=0 THEN GOTO 1190
1170 PRINT N$(80);N$(26)
1180 GOSUB 2030
1190 CLS
1200 PRINT "IMAGINATE QUE TE ENCUENTRAS EN UNA ISLA DESIERTA DESPUES DE UN NAUFR
      AG10 Y QUE TIENES UNICAMENTE PARA HACER SEÑALES A UN BARCO QUE PASA, UN Q
      UINQUE; PERO CON EL PROBLEMA DE QUE EL ACEITE QUE TIENES PARA ENCENDERLO NO
      LLEGA "
1210 PRINT "A LA MECHA PORQUE ES Poca CANTIDAD Y NO TIENES MAS COSAS QUE UNAS CE
      RILLAS ¿COMO TE LAS ARREGLARIAS PARA ENCENDER EL QUINQUE?"
1220 GOSUB 2000
1230 IF RES=0 THEN GOTO 1260
1240 PRINT N$(61);N$(39);N$(44);N$(42);N$(69);N$(82);N$(51);N$(63);N$(49);N$(24)
1250 GOSUB 2030
1260 CLS
1270 PRINT "¿QUE RIO ESPAÑOL DESPUES DE NACER DESAPARECE Y VUELVE A APARECER MAS
      ADELANTE?"
1280 GOSUB 2000
1290 IF RES=0 THEN GOTO 1320
1300 PRINT N$(62);N$(2)
1310 GOSUB 2030
1320 CLS
1330 PRINT "¿CUAL ES EL SIGUIENTE NUMERO DE ESTA SERIE:?"
1340 PRINT "1-4-9-16-25"
1350 GOSUB 2000
1360 IF RES=0 THEN GOTO 1390
1370 PRINT N$(5);N$(12)
1380 GOSUB 2030
1390 CLS:PRINT "YA NO HAY MAS PREGUNTAS"
1400 END
2000 LOCATE 15,1
2010 PRINT "PULSAR LAS TECLAS (MAYUSCULAS):"
2020 PRINT "R-->CONOCER LA RESPUESTA":GOTO 2040
2030 PRINT
2040 PRINT "O-->OTRA PREGUNTA"
2050 PRINT "T-->TERMINAR "
2060 LET A$=INKEY$
2070 IF A$="" THEN GOTO 2060
2080 IF A$="T" THEN END
2090 IF A$="R" THEN LET RES=1:RETURN
2100 IF A$="O" THEN LET RES=0:RETURN
2110 GOTO 2040
2120 REM *****
2130 REM ** MODIFICACIONES PARA SPECTRUM: **
2140 REM ** 60 DIM N$(85,35) **
2150 REM ** 2000 PRINT AT 8,1 **
2160 REM ** 2080 IF A$="T" THEN GOTO 9999 **
2170 REM ** MODIFICACIONES PARA COMMODORE: **
2180 REM ** Sustituir todos los CLS por PRINT CHR$(147) **
2190 REM ** 2000 PRINT :PRINT:PRINT **
2200 REM ** 2060 GET A$ **
2210 REM *****

```

Explicación del programa

La estructura del programa es muy sencilla. Comienza con una serie de instrucciones

DATA que son donde se encuentran las respuestas a las preguntas. Estas respuestas están puestas en estos DATA partidas y de ma-

nera desordenada para que al mirar el listado o teclearlo en el ordenador no se puedan conocer éstas, ya que entonces el juego perdería su interés. En total, las 18 respuestas se han dividido en 85 partes y a continuación de los DATA (líneas 250-270) se leen todas ellas y se guardan en la lista N\$.

A partir de la instrucción 280 el programa comienza a escribir las distintas preguntas y para todas ellas tiene la misma estructura:

- Borra la pantalla y escribe el texto de la pregunta.

- Hace una llamada a la subrutina 2000: esta subrutina escribe las tres opciones con las teclas que hay que pulsar para cada una de ellas y, en caso de que se solicite la respuesta, antes de terminar pone un 1 en la variable RES. Si se solicita la siguiente pregunta, directamente pone un cero en esta variable y si se pide terminar, termina directamente.

- Se comprueba si en la variable RES hay un cero y, si es así, se salta a la línea donde comienza la siguiente pregunta.

- Si no se ha saltado, se escribe la respuesta, que estará en las variables correspondientes de la lista N\$, escribiéndose en cada pregunta aquellas variables en las que se encuentren las partes correspondientes a su respuesta debidamente ordenadas.

- Se llama a la subrutina 2030, que hace lo mismo que la 2000, con la excepción de que no ofrece la opción de conocer la respuesta que ahora ya no se necesita.

NOTA: Se aconseja tener especial cuidado a la hora de teclear los datos correspondientes a las instrucciones DATA, ya que una coma o unas comillas omitidas o cambiadas de lugar harían que el programa diese luego todas las respuestas de manera errónea.

PEQUEÑA HISTORIA DE LA INFORMATICA

■ El MARK I, ¿primer ordenador?



GEORGE Stibitz, de los laboratorios Bell, no fue el único que construyó calculadoras digitales utilizando relés como interruptores. En 1937, un joven ingeniero, Claude E. Shannon aprovechó su tesis de licenciatura para es-

tudiar una aplicación práctica: la utilización de los números binarios y la Lógica simbólica en el diseño de calculadores. Ni Shannon ni Stibitz conocían cada uno de los trabajos del otro, y sin embargo, sus hallazgos fueron parejos. Pero sus destinos los iban a unir, años más tarde Shannon pasó a trabajar en el equipo de ingenieros de los laboratorios Bell, que en esos años estaba a la cabeza de la tecnología punta. La colaboración de Shannon, fue sobre todo en el aspecto matemático, publicando un trabajo bastante completo sobre las comunicaciones describiéndolas mediante algoritmos matemáticos. Su trabajo apareció en el Bell System Technical Journal, donde él escribe también por primera vez la designación de «Bit» como contracción para referirse a un dígito binario. Su artículo apareció en el año 1948, en dicha publicación de la Compañía Bell, once años más tarde del día en que comenzó sus primeros trabajos con los calculadores de relés.

Algunos años antes de la publicación de la teoría matemática de Shannon, en 1944, Howard Aiken comenzó a interesarse en la fabricación de un calculador universal, un proyecto que se asemejaba en cierto modo al calculador soñado por Babbage, y que éste nunca había conseguido llevar a la práctica. Howard

Aiken, se encontraba por aquel entonces en la Universidad de Harvard, y también como Shannon estaba trabajando en el desarrollo de su tesis doctoral en Ciencias Físicas. Como los aspectos teóricos de su tesis implicaban encontrar la solución de muchas ecuaciones diferenciales no lineales, y el sistema utilizado para resolverlas era el de aproximaciones numéricas, el problema se eternizaba, y se consumían tiempos que podían aprovecharse más fructíferamente.

Howard Aiken comenzó a considerar el diseño de una máquina que le resolviera estos cálculos pesadísimos, y se puso en marcha. En no mucho tiempo, ya había diseñado e incluso desarrollado una máquina que calculaba polinomios sencillos. Sus trabajos en ese tipo de máquinas continuaron aún varios años, sin que Aiken se desengañara, pero, finalmente, consideró que no había mucho más que perfeccionar en este tipo de máquinas, y, aunque naturalmente cada modelo era generalmente una mejora del anterior, en realidad, eran máquinas sin demasiadas posibilidades de futuro. Durante el tiempo que pasó intentando obtener la «herramienta» necesaria para ayudarle en su trabajo, Aiken se fue interesando más y más en el tema de las máquinas automáticas, hasta el punto de que se decidió a diseñar una, que fuera capaz de tratar cualquier tipo de problema, es decir, una máquina de uso general.

Thomas Watson senior, siempre muy introducido en todos los ambientes de élite, conoció a Howard Aiken en 1939, y quedó muy impresionado con sus trabajos, y sus ideas para el futuro de los calculadores. Le prometió apoyo económico, e incluso colaboró, enviándole personal especializado de su empre-

sa para que le ayudara en sus trabajos. Entre dichos ingenieros de IBM se encontraban B. M. Durfee, F. E. Hamilton, C. D. Lake y J. W. Bryce, todos hombres con mucho prestigio.

La máquina se llamó Calculador Automático de Secuencias Controlado, o Harvard Mark 1, porque había sido desarrollado en la Universidad de Harvard. El Mark 1, era electromecánico, es decir, las operaciones se llevaban a cabo por elementos mecánicos, controlados eléctricamente. Llevaba multitud de relés, y era una máquina realmente enorme. Además, podía tomar ciertas decisiones, lo que la convertía en una máquina universal, es decir, *el primer ordenador de la historia*.

Es interesante notar que Aiken no conocía los trabajos de Babbage, y llegó a estudiarlos tres años después de comenzar sus trabajos en el Mark 1. En uno de sus escritos, Aiken llega a comentar, no sin cierta sorna: «Si Babbage llega a nacer setenta y cinco años más tarde, me hubiera dejado sin trabajo.» Efectivamente, Charles Babbage había dedicado su vida al mismo objetivo que Aiken, aunque los medios, las circunstancias de su época, le impidieron lograrlos.

Digamos algo sobre el Mark 1, aunque sea del dominio general. Era una máquina muy grande, que medía más de 15 metros de longitud por veinticuatro metros. Estaba formado por 760.000 elementos unidos por más de 800 kilómetros de cables. Naturalmente, realizaba las operaciones matemáticas elementales, y además, en otros módulos, calculaba senos, logaritmos, y otras funciones matemáticas.

La enorme máquina, había sido diseñada para que realizara todas las operaciones de forma totalmente automática, es decir, sin colaboración humana de ningún tipo. Las instrucciones a la máquina, debía darlas una persona especializada (programador), que las grababa en una cinta de papel, perforándolo adecuadamente. Las perforaciones se leían electrónicamente fila tras fila, teniendo cada fila veinticuatro posiciones para posibles perforaciones. Los números sobre los que debían operar las instrucciones se almacenaban en registros. Estas instrucciones, naturalmente, eran absolutamente elementales, del tipo de «Toma lo que está en el registro 29 y añádelo a lo que existe en el registro 63, continuando la lectura de la línea siguiente». En las instrucciones, las direcciones de los registros, y los códigos de las acciones a realizar aparecen unas tras las



Grace Hooper, colaboradora de Howard Aiken.

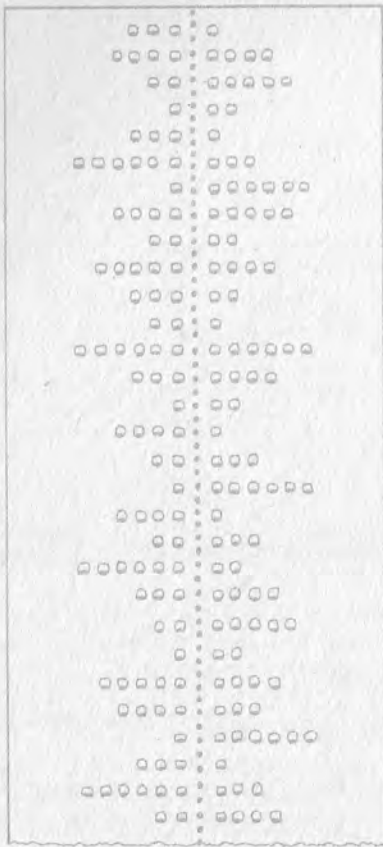
otras. Los cálculos se van descomponiendo en pequeños pasos sucesivamente, de forma que para realizar un cálculo cualquiera, la «programación» era larguísima, y debía realizarse exquisitamente, y por personas muy ordenadas y sistemáticas.

El Hardware de la máquina estaba compuesto por registros formados por juegos de ruedas. Cada rueda (llamada rueda de contador) disponía de diez posiciones, pudiendo por tanto almacenar un dígito decimal. Cada registro de almacenamiento estaba compuesto por veinticuatro ruedas (veintitrés decimales más el signo). El número total de registros era de setenta y dos, ¡una cifra enorme!

Una característica importante, es que cada registro de almacenamiento disponía de un mecanismo sumador completo, de forma que se pudiese sumar (o restar) directamente a otro. (En realidad, cada registro era simplemente un acumulador, es decir, podía almacenar datos, y realizar operaciones elementales de suma y resta.) Tanto la multiplicación como la división se realizaban en otros módulos separados. Las constantes se podían almacenar manualmente, accionando los correspondientes interruptores. El número total de constantes que se podían guardar era de sesenta.

Cada grupo de ruedas de contador estaba asociada a un eje de rotación continua. Cualquiera de las ruedas de contador se po-

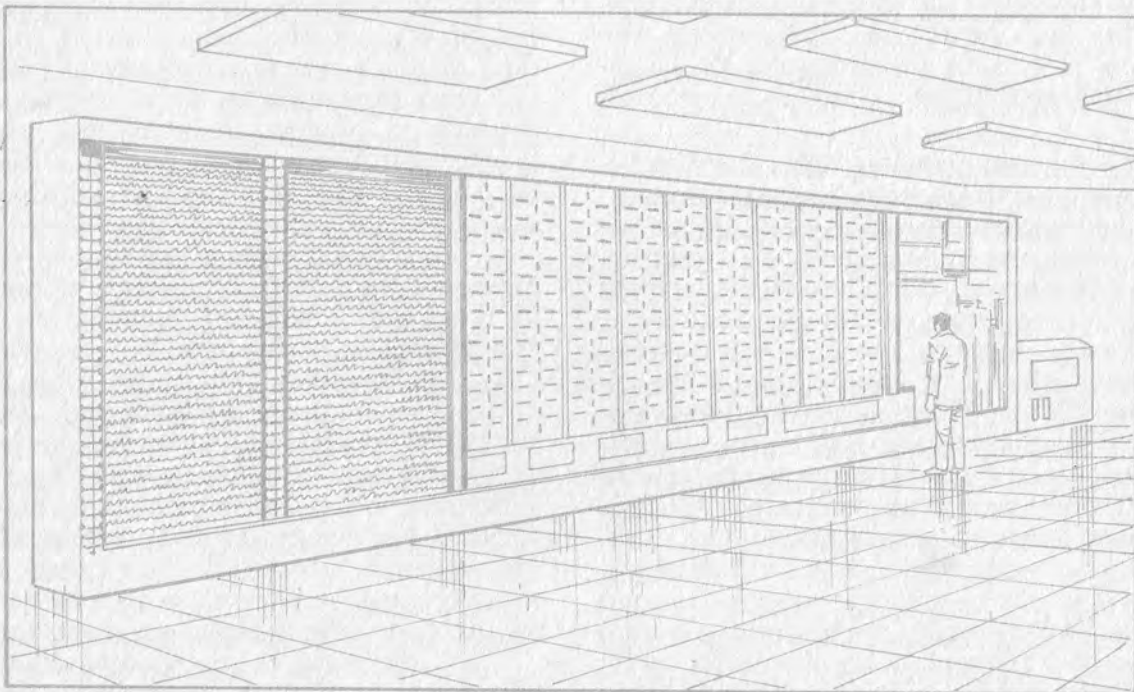
PEQUEÑA HISTORIA DE LA INFORMATICA



Cinta codificada.

día conectar a ese eje, para cualquier fracción de revolución, y el ensamblaje se realizaba mediante un embrague electromagnético controlado por un relé normal, de los utilizados en las centrales telefónicas automáticas. Cada contador disponía de una pequeña escobilla, que podía actuar sobre diez contactos, indicando así eléctricamente su posición. La suma de un número a otro se realizaba haciendo las correspondientes conexiones con las escobillas de un acumulador A, y con el mecanismo de embrague de otro acumulador, en este caso, B, girando sus ruedas de acuerdo con el número almacenado en el acumulador A.

El Mark 1 disponía de unos tres mil relés. El tipo de relés que utilizaba era muy sencillo, consistiendo simplemente en una barra pequeña metálica, unida a un muelle. Por la acción de un electroimán, la barra se puede levantar. Cuando el imán se desactiva, la barra cae, cerrando el circuito correspondiente. Como puede deducirse de la explicación anterior, los relés de que hablamos eran bastante lentos, y desde luego enormes, y pesados (cada uno de ellos medía unos tres centímetros, pero no debe olvidarse el gran número necesario, y además todo el resto de componentes). Como la velocidad de apertura, cerrado de cada relé era del orden de una centésima de segundo, las velocidades (aun-



El Mark I.

que muy buenas en su momento) eran lentísimas. El número mayor de cifras admisible para los cálculos numéricos era veintitrés, y, por ejemplo para realizar una multiplicación de dos números de veintitrés cifras, tardaba unos cuatro segundos y medio. Para sumas y restas de números, el tiempo era mucho menor: 0,3 segundos.

La memoria de la máquina disponía de 72 posiciones decimales, estando formada cada una de ellas por veinticuatro ruedas de contador de diez posiciones, una para cada cifra decimal. Además, también tenía otras sesenta y nueve posiciones adicionales, para constantes, de las que ya hemos hablado.

En cuanto a los elementos de entrada y salida, también eran bastante completos. La máquina se había construido sin reparar en gastos, ni espacio, ni otros condicionamientos «barriles». ¡Las obras de arte no tienen precio! El equipo disponía además de dos lectoras de tarjetas, una perforadora y dos máquinas de escribir. También podían utilizarse unos mecanismos especiales directos, para hallar los valores numéricos de las funciones más corrientes, que incluso se presentaban al usuario en cintas especiales. En cuanto a los datos, se introducían por tarjetas, y podían extraerse con el mismo formato, o bien mecanografiados.

Hasta ahora no hemos hablado del ruido. Bien es cierto que no disponía del BEEP de los pequeños ordenadores modernos, sin embargo, el tremendo repiqueteo de los relés, producido durante el funcionamiento de la máquina, recordaba un taller de costura, lleno de señoras trabajando en las máquinas.

Parece ser que en la presentación de la nueva máquina, Watson apenas si fue citado, a pesar de haber contribuido con dinero y

hombres a su creación, considerando la generalidad de los asistentes a dicho acontecimiento la máquina como propiedad de la Universidad. La realidad es que las patentes pertenecían a Watson, aunque éste las cediera a la Universidad. A partir de este momento, las relaciones de Aiken con Watson nunca volvieron a ser buenas, ya que Aiken consideró siempre a la máquina como suya, y Watson era demasiado importante para soportar orgullos ajenos. Aunque el prototipo permaneció en la Universidad durante muchos años (más de quince) en que fue utilizado sin descanso, sobre todo por la Marina de los Estados Unidos, Watson desarrolló rápidamente en su empresa una nueva máquina para sus propias necesidades.

Grace Hooper, colaboradora de Howard Aiken desde los primeros momentos, escribió cómo se realizó la primera «depuración» del incipiente ordenador. En uno de sus frecuentes fallos, el equipo destinado a subsanarlo, compuesto de numerosas personas, tardó varias horas en descubrir el problema que había producido el error. Se trataba simplemente de un pequeño bichito que se encontraba en uno de los relés, impidiendo su correcto funcionamiento. El insecto fue sacado de su impertinente posición, y guardado en el libro de instrucciones, como «mascota», ya que constituía la primera «depuración» del enorme equipo.

En 1947 se construyó otra máquina también de relés, pero más rápida, a la que se llamó Mark II.

Sin embargo, las máquinas de relés eran muy lentas, complicadas y pesadas (todo ello debido a sus partes mecánicas). Con el tiempo, los relés fueron sustituidos por tubos de vacío.

■ Sistemas de gestión de base de datos con ordenadores backend



CONTINUANDO el tema tratado anteriormente, y en contraposición a los sistemas DBC (máquinas hardware de bases de datos) ya descritos, vamos a considerar la arquitectura específica del software de un sistema de gestión de bases de datos, con ordenadores distribuidos (MBDS, Multi Backend Database System) soportado en ordenadores de propósito general.

En la figura aparece representada la estructura general de uno de dichos sistemas, donde se ven los diferentes ordenadores (backend) de manejo de los discos en que se soporta la base de datos propiamente dicha, el controlador y el ordenador principal (host) donde se generan las consultas.

El esquema general de funcionamiento del sistema, es el siguiente:

Cuando se recibe una transacción desde el ordenador principal, o desde uno de los terminales, el controlador envía (difunde) dicha transacción en todos los ordenadores backend. Cada backend dispone de cierto número de unidades de disco dedicadas. Como todos los datos están distribuidos entre los backend, la transacción se puede ejecutar simultáneamente en todos los ordenadores backend, manteniendo cada uno de ellos una cola independiente, así como esquemas de tiempo no relacionados, todo ello con el fin de mejorar cuanto sea posible las operaciones de acceso, y disminuir los tiempos muertos. Con esta técnica, las operaciones se ejecutan en los backends de forma concurrente. El controlador no tiene, por tanto, una función muy larga ni com-

plicada. Sin embargo, se ocupa de la emisión-difusión, de encaminar adecuadamente las transacciones y auxiliar en la inserción de nuevos datos. Los backends realizan todas las operaciones primarias de la base de datos.

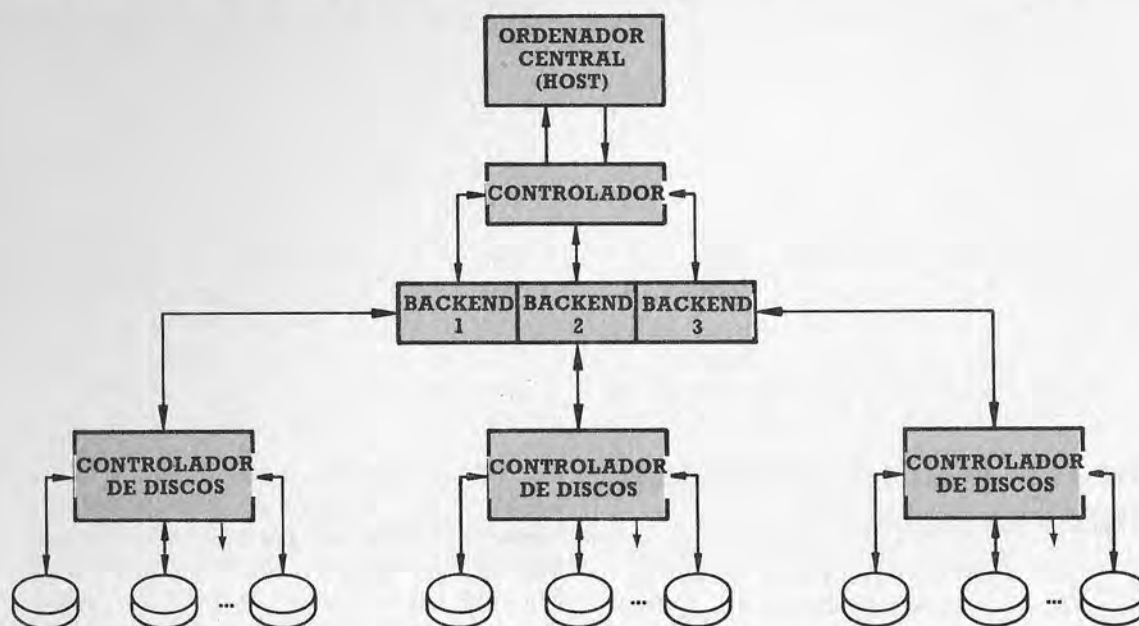
Las comunicaciones entre ordenadores, en MBDS se realizan utilizando un bus multiplexado de división de tiempos, al que se denomina unión de comunicaciones paralela PCL. El MBDS proporciona una representación abstracta por software de dicho bus, para cada ordenador, con una finalidad muy concreta: emular la capacidad de difusión de transacciones. Dicha representación abstracta consiste en dos procesos complementarios: El primero de ellos, get-pcl, capta mensajes desde otros ordenadores externos al PCL. El segundo proceso, put-pcl, envía mensajes al bus, para que éste los distribuya a los otros ordenadores.

Cada uno de los ordenadores, tanto el controlador como uno cualquiera de los que forman parte del grupo backend, disponen de sus propios procesos, get-pcl y put-pcl.

El conjunto MBDS debe poder ser expandible, y capaz de autoconfigurarse para que el sistema disponga de un número cualquiera de backends. Además, el hardware y el software de los ordenadores backend puede ser idéntico, permitiendo añadir más ordenadores al sistema sin más que repetir una copia del software y adquirir ordenadores idénticos a los utilizados (idéntico hardware).

En los sistemas de gestión de bases de datos, que disponen de multi-backend, existen dos objetivos básicos:

— El primero es incrementar el número de backends, manteniendo el tamaño de la base de datos, número y formato de las respuestas a las transacciones. Con ello se lograría una reducción de los tiempos de respues-



ESTRUCTURA DE UN SISTEMA MBDS

ta de las transacciones realizadas por los usuarios.

— El segundo objetivo es incrementar el número de ordenadores backend de forma proporcional al incremento de las respuestas a transacciones.

Una vez indicada la estructura del sistema (hardware + software) multi-backend, podemos revisar la estructura de la base de datos propiamente dicha, que se suele establecer según un modelo de datos «basado en los atributos». En este tipo de bases de datos, los datos están organizados considerando jerárquicamente los siguientes elementos: base de datos, archivo, registro, par de valores correspondientes a atributos, palabra clave, gama de valores para los atributos, palabra clave de directorio, palabra clave no de directorio, directorio, cuerpo del registro, predicado de la palabra clave y consulta.

Expliquémoslo de forma sencilla: una base de datos es, a nivel elemental, una colección de archivos. Cada archivo está formado por un conjunto de registros, caracterizados por un conjunto único de palabras clave de directorio. A su vez, el registro está formado por dos partes fundamentales: una de ellas es un grupo de pares de valor-atributo, o palabras clave. Los pares valor-atributo son elementos del producto cartesiano del nombre del atributo, y un valor dentro del dominio de valo-

res posibles, para el atributo. Veamos un ejemplo, (VIAJES, 300), es un par valor-atributo, siendo 300 el valor del atributo VIAJES. Los registros, por otro lado, contienen por lo menos un par valor-atributo, por cada atributo definido en la base de datos. Algunos pares valor-atributo de un registro (o de un fichero) se llaman palabras clave a dicho registro (o archivo), ya que en el directorio (y con fines identificativos) aparecen los pares valor-atributo, o la gama de dichos valores de atributo.

Todos los pares valor-atributo que no aparezcan en el directorio, son los llamados palabras clave de no directorio.

La otra parte del registro, está compuesta por texto, información que se denomina «cuerpo del registro».

Veamos un ejemplo:

```

(<FILE, Libros>, <TIPO, Ficción>,
<AUTOR, Delibes>, {Aventuras, Encuaderna
do})

```

Normalmente, los directorios se estructuran a tres niveles: atributos, descriptores, y agrupaciones («clusters»). Los valores de estos elementos se agrupan en tres tablas. En la primera tabla, (tabla de atributos o AT), se reservan los diferentes atributos con la referencia a la posición que en la segunda tabla tiene su descripción. Esta otra tabla (tabla de descriptores o DDIT) especifica el rango de valores

admisibles para cada atributo y el identificador correspondiente a cada uno de los diferentes intervalos admisibles dentro el rango de valores. La tabla de definición de agrupamientos (CDT) describe la manera en que se agrupan los identificadores que aparecen en tabla DDIT y su ubicación en la base de datos.

Veamos a continuación cómo se distribuyen los datos en el sistema multi-backend. Disponemos de dos tipos de datos, datos de directorio, y datos de usuario. En cuanto a los primeros, el camino a seguir consiste en realizar copias de dichos datos de directorio para cada uno de los ordenadores backend. El segundo tipo de datos, por el contrario, se distribuye más o menos equitativamente entre los sistemas backend.

Por tanto, cada sistema backend dispondrá de su propia tabla de directorios. Las tablas, son siempre iguales, pero no incluyen los identificadores de registro, es decir, los identificadores de la tabla CDT no son los mismos en cada equipo backend, pero son únicos para un backend dado, pues son las direcciones secundarias de los registros almacenados en los discos de los backends. Las tablas de directorios se almacenan también en un almacenamiento secundario, y pasarán a la memoria primaria, en el momento en que sean necesarias.

Cuando las consultas constan de muchos predicados, es muy útil utilizar las copias de los directorios, poniendo a trabajar un backend en un conjunto de predicados distinto de los demás, para a continuación, intercambiar resultados entre los sistemas backend. Con este procedimiento, se realizan búsquedas en paralelo en los directorios, para distintos predicados.

Cuando las consultas constan de un número pequeño de predicados, la consulta completa se envía a cada uno de los sistemas backends, que realizan la búsqueda en el directorio en paralelo para la consulta total.

Normalmente, el espacio que toman los directorios dentro de la base de datos supone un porcentaje considerable de la propia base de datos. (de un 10 a un 20 %). Sin embargo, al realizar las copias del directorio, se pueden conseguir mejores (menores) tiempos de acceso a los datos beneficiando de este modo al usuario, al que se limita el espacio de búsqueda, dentro de la base de datos. El inconveniente correspondiente, es que en cada uno de los sistemas debe encontrarse una copia de los directorios. Sin embargo, en la práctica se ha

visto que las ventajas de esta estrategia son mayores que los inconvenientes.

La información de los directorios, puede ser tan actualizada dinámicamente durante la ejecución de las consultas, sin que existan complicaciones especiales debidas a estos directorios múltiples.

La ubicación de la base de datos de estructura de agrupamientos es tarea típica del controlador, aunque se lleva a cabo por los backends. Estos, forman nuevos agrupamientos (clusters), y cuando sea necesario incluir un nuevo registro en estos agrupamientos, el controlador es el dispositivo que debe decidir cuál de los ordenadores backend insertará el registro en el cluster correspondiente.

La inserción del registro en el agrupamiento se lleva a cabo colocando éste en el bloque de almacenamiento secundario del sistema backend seleccionado, y así continuará la operación (dirigida por el controlador) introduciendo registros en el mismo grupo del bloque, hasta que dicho bloque de almacenamiento secundario se llene por completo. Cuando esto ocurra, el controlador hará que otro backend continúe la operación, colocando nuevos registros dentro del mismo agrupamiento (cluster). El controlador siempre tiene información sobre aquellos sistemas backend cuyos almacenamientos secundarios pueden ser usados para la inserción de nuevos registros en los clusters existentes.

Veamos someramente el controlador. Realiza tres procesos: preparación de la consulta, generación de la información a insertar, y proceso posterior (posproceso). La primera de estas actividades consiste en recibir, interpretar y dar formato a la consulta (transacción), antes de enviarla a cada backend para que se efectúe el proceso de gestión de directorio (del que hemos hablado anteriormente). El proceso siguiente, de generación de la información a insertar, se utiliza para proporcionar a los backends información adicional en los casos en los que la transacción requiera inserciones. En este caso, la inserción sólo se lleva a cabo en uno (único) de los sistemas backend, ya que la información del usuario no está repetida, y se encuentra repartida entre los backends. El tercer proceso se utiliza para recoger los resultados de la transacción y presentarlos al usuario.

La tarea del controlador es importante, así como todo el conjunto de comunicaciones entre los ordenadores que componen el sistema que gestiona la base de datos. No hay que

olvidar que un controlador puede procesar unas 3.000 líneas de código C (incluyendo dentro de ello los procesos de comunicaciones), ocupando casi 100 KBytes de código ejecutable.

A su vez, los backends realizan otros tres procesos diferentes: buscar en las tablas de directorios (gestión de directorios), asegurar la consistencia del directorio de datos, y de los datos del usuario (control de concurrencia) el control de las operaciones con los discos (E/S) y otras operaciones especificadas por la propia consulta.

El software de los equipos backend es completo, y, por tanto, puede realizar todas las operaciones de la base de datos primaria. El software instalado en cada backend consiste en unas 12.000 líneas de código C, con un código ejecutable de 250 Kbytes.

De hecho, los backends sólo se comunican con el controlador para dirigir los resultados de la transacción, o para la inserción de registros. En cuanto a las comunicaciones con otros equipos backend, esta sólo existe durante una fase específica de la búsqueda en los directorios.

TERMINOLOGIA

AT. Siglas de Attribute Table, Tabla de Atributos. Tabla incluida en los directorios de una base de datos, donde se incluyen todos los atributos a utilizar en la base de datos, junto con la referencia del punto en que sus valores vienen referenciados en la DDIT.

Atributo. Cada una de las características diferenciales o parámetros representativos de un registro frente a los restantes de la base de datos.

Backend, procesador. Procesador diseñado para una tarea específica. Se llama en ocasiones, también, procesador dedicado, o de una sola tarea. Suele estar especializado para la realización básica de operaciones aritméticas, para la gestión de algunos periféricos específicos, o el manejo de bases de datos.

Bus. Canal común de comunicaciones, compartido por varios ordenadores y/o periféricos.

CDT. Tabla de definición de agrupamientos (cluster definition table) en la que se describe cómo se agrupan los identificadores de la tabla DDIT y su situación dentro del conjunto de los dispositivos donde está soportada la base de datos.

Cluster. Agrupación de elementos. En las bases de datos estructuradas por atributos, conjunto de registros que se agrupan como una unidad, para su tratamiento por el software del sistema. Es usual utilizar el término cluster también para el agrupamiento de un conjunto de unidades físicas, cintas magnéticas, discos, etc., cuando éstas tienen uno o varios caminos comunes de acceso.

Concurrente, ejecución. Ejecución simultánea (paralela) de varios subprocesos o tareas parciales dentro de un proceso complejo. En las bases de datos distribuidas y controladas por varios procesadores backend, se puede realizar una ejecución paralela de accesos para la búsqueda de la información.

DBC. Procesador de Bases de Datos (Database Computer). Procesador dedicado a la gestión de bases de datos, por lo que dispone de una estructura especialmente apta para el manejo de las direcciones de los datos y para realizar accesos en paralelo, agrupamiento de datos para su lectura y/o escritura, etc.

DDIT. Tabla de descriptores. Descriptor-to-Descriptor-Id Table. Relación incluida en los índices de la Base de Datos, para especificar el rango de los valores admisibles para cada atributo y los identificadores correspondientes.

Difusión. Diseminación o distribución no selectiva de la información. En una red de comunicaciones o un sistema de manejo de la información en general, se designa con este término al envío de un mensaje o de una información a todos los destinatarios posibles, sin realizar ninguna selección. En contraposición a transmisión selectiva, o dirigida.

MBDS. Multi-backend Database System. Sistema de Base de datos con ordenadores backend. Sistema basado en un red de ordenadores (backend) que gobiernan, cada uno de ellos, un grupo de dispositivos físicos de almacenamiento de datos.

TERMINOLOGIA

PCL. Parallel Communication Link. Canal de unión para comunicaciones en paralelo. Es el tipo de bus utilizado usualmente en los MBDS para la comunicación entre los distintos ordenadores backend, el controlador y el ordenador central.

Posproceso. Elaboración de la información que realiza el controlador una vez concluida la consulta a la Base de Datos, para la

adecuada presentación al usuario de la respuesta.

Transacción. Tarea elemental que ha de realizar el sistema. Normalmente se alude con este término al proceso de un solo registro, al envío de un mensaje simple en una red de comunicaciones, o, en una MBDS, a la secuencia siguiente: elaboración de la pregunta, acceso a la Base de Datos y preparación de la respuesta.

VOCABULARIO DE INFORMATICA

Menor o igual que. Operador de relación, válido también en casos de igualdad.

Menor que. Operador de relación, representado por el símbolo <.

Mensaje. Información cuyo comienzo y final están adecuadamente definidos, o bien se conocen de forma implícita. Conjunto de datos transmitidos por un canal de comunicaciones.

Menú. Lista de procedimientos o programas, entre los que el usuario puede optar para su ejecución. También, relación de opciones a elegir (como valor de un parámetro) en un proceso.

Merge (intercalación). Término inglés que indica una operación por la cual se combinan elementos de dos conjuntos ordenados con el mismo criterio, para formar otro conjunto, que también cumple las mismas normas en cuanto a la organización de sus elementos.

Metalenguaje. Lenguaje utilizado para referirse a otro lenguaje.

Micro. Prefijo multiplicador, que indica una millonésima parte de algo (μ). Este término se ha impuesto como prefijo de numerosas palabras para indicar su pequeño tamaño. Así, tenemos microordenador, microcircuito, microprocesador, etc.

Microinstrucción. En lenguaje máquina, algunas instrucciones se pueden dividir a su vez en otras instrucciones muy elementales, llamadas microinstrucciones. Es un sistema para definir distintos grupos de instrucciones, flexibilizando la unidad de control del ordenador.

Microordenador. Ordenador muy pequeño, o cuya CPU es un microprocesador. Se utiliza hoy en día como sinónimo de ordenador personal u ordenador profesional.

Microprogramación. Programación con microinstrucciones.

Microprocesador. Componente electrónico construido con tecnología de integración a gran escala, y que contiene las partes fundamentales de un ordenador (unidad de control), unidad lógico aritmética y registros de uso general). Si se incluye al microprocesador una memoria, y un sistema de entrada y salida de datos, se habrá completado un microordenador. Microprocesadores muy utilizados son el Z80, 8086/8088, 68000, 6800, etc.

Microprocesor. (Ver **Microprocesador**).

Microsecond. (Ver **Microsegundo**).

Microsegundo. Unidad de tiempo muy utilizada en informática. Millonésima parte de un segundo.

Milisecond. (Ver **Milisegundo**).

Milisegundo. Milésima parte de un segundo.

Miniaturización. Técnica electrónica de construcción del hardware de los equipos que persigue la utilización de componentes con dimensiones cada vez menores, para obtener mejores características en los sistemas.

Miniordenador. Ordenador de dimensiones medias (en comparación con los ordenadores de grandes dimensiones, o mainframes). Fueron los primeros ordenadores pequeños

VOCABULARIO DE INFORMATICA

que aparecieron en el mercado, aunque algo más tarde llegaron los ordenadores personales muy pequeños, dejando obsoleto este término.

MIPS. Unidad de medida de la velocidad de proceso de un ordenador. Las siglas millón de instrucciones por segundo indican el sistema de medida utilizado. Se utiliza básicamente para medir la capacidad de proceso de los ordenadores medios y grandes.

MIS. Siglas de Management Information System. Sistema de información de la dirección. Se llaman así los sistemas integrados de gestión de la información, en un entorno empresarial.

Mistake. Término inglés que significa error. En informática suele referirse a un error humano, que produce un resultado no esperado.

Mnemónico. Nombre simbólico que recuerda el significado del dato al que se refiere. Tiene el mismo significado que nemotécnico. Puede ser una palabra, un símbolo, una contracción o un apócope, como defect, para indicar defectuoso.

Modem. Contracción de Modulador-Demodulador. Dispositivo que modula señales, es decir, las prepara para una mejor transmisión a través de un medio físico, siendo capaz, además, de realizar la operación inversa, y demodular la señal en el extremo opuesto del canal de transmisión.

Modo. Véase el tipo de modo, de programación, de direccionamiento, de carga, etc.

Modo de carga. En algunos ordenadores que aceptan longitudes de palabras variables, transmisión de los datos, en la que los delimitadores se mueven con los propios datos.

Modo de direccionamiento. Uno de los métodos de acceso a una posición de memoria utilizando lenguaje máquina o ensamblador. Los modos de direccionamiento más usuales son directo, indirecto, indexado, inmediato y relativo.

Modo de programación. Método de ejecución de los programas. Existen dos tipos fundamentales: directo, cuando las instrucciones se van ejecutando una a una nada más introducirlas en el ordenador, e indirecto o diferido, en el que las instrucciones están numeradas y almacenadas, y se ejecutan una tras otra sin más intervención humana que el envío del comando correspondiente.

Módulo. Parte de un programa que en sí mismo constituye una unidad en la compilación, carga y otra faceta. También puede referirse a un paquete de hardware que se puede integrar junto con otros componentes. Según otra acepción, puede referirse al resto de la división entre dos números. Por ejemplo, $9 \text{ MOD } 4 = 1$.

Monádico, operador booleano. Operador booleano que sólo tiene un operando, por ejemplo, NOT.

Monádica, operación. Operación sobre un operante único, por ejemplo, la negación.

Monitor. Elemento de software o hardware que observa, supervisa, controla o verifica la operación de un sistema. También puede referirse a una unidad de presentación visual, muy semejante a un televisor doméstico en muchos aspectos (no lleva el sistema de recepción de señales de la antena). La calidad de la presentación en los monitores (texto y gráficos) suele ser superior a la de los televisores.

